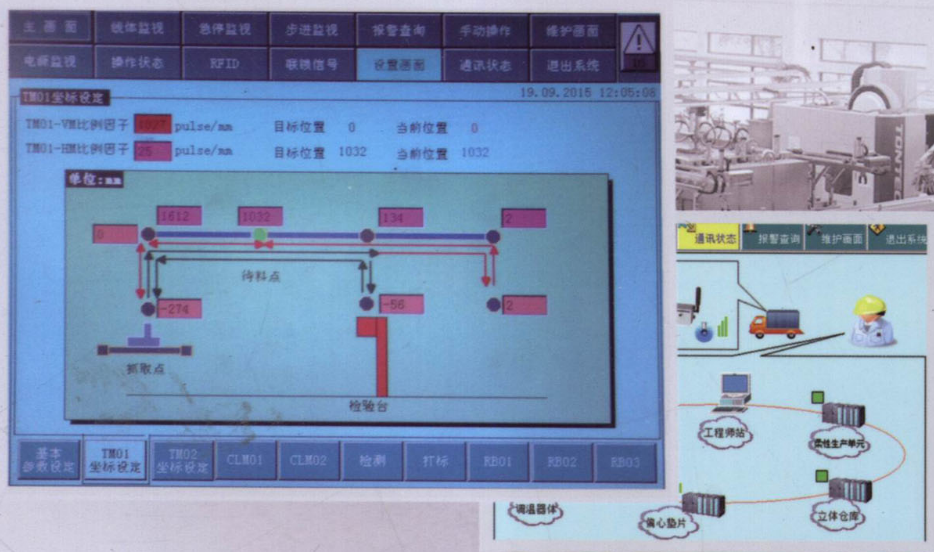
电 气 信 息 工 程 丛 书

S7-200/S7-300 PLC 基础及系统集成

姚晓宁 郭 琼 主编





本书配有相关案例源代码

下载网址为 www.cmpbook.com





机械工业出版社
CHINA MACHINE PRESS

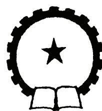
VISIT…

LANZAROTE Caliente.com

电气信息工程丛书

S7-200/S7-300 PLC 基础及系统集成

主　编　姚晓宁　郭　琼
副主编　刘志刚
参　编　冯　笠　秦　爽　谢　端



机 械 工 业 出 版 社

本书以 S7-200 PLC、S7-300 PLC 为基础，采用案例带动知识点，全面细致地讲解了中小型 PLC 基础指令、程序编写、通信方式及系统设计与实现，并围绕 S7-200 PLC、S7-300 PLC 讲述了 PLC 及其关联设备的系统集成概念、系统集成方法及系统集成案例。

本书在内容安排上强调 PLC 及其系统集成的实际应用，紧密结合控制技术的新发展、新应用，案例丰富翔实，结构合理，可作为高校自动化类专业的教材，也可作为从事 PLC、现场总线、系统集成等设计与应用开发技术人员的培训教材或参考资料。

图书在版编目（CIP）数据

S7-200/S7-300 PLC 基础及系统集成/姚晓宁，郭琼主编. -北京：机械工业出版社，2015. 8
（电气信息工程丛书）
ISBN 978-7-111-51420-6

Ⅰ. ①S… Ⅱ. ①姚… ②郭… Ⅲ. ①plc 技术 Ⅳ. ①TM571. 6

中国版本图书馆 CIP 数据核字（2015）第 206551 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）
策划编辑：时 静 责任编辑：时 静
责任校对：张艳霞 责任印制：李 洋
北京振兴源印务有限公司印刷
2015 年 10 月第 1 版 · 第 1 次印刷
184mm×260mm · 21. 75印张 · 537 千字
0001—3000 册
标准书号：ISBN 978-7-111-51420-6
定价：59. 80 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务	网络服务
服务咨询热线：010-88361066	机 工 官 网：www. cmpbook. com
读者购书热线：010-68326294	机 工 官 博：weibo. com/cmp1952
010-88379203	金 书 网：www. golden-book. com
封面无防伪标均为盗版	教育服务网：www. cmpedu. com

前　言

可编程序控制器（PLC）是一种以微型计算机为核心的通用工业控制器。从其产生发展到现在，其控制功能和应用领域不断拓展，实现了由单体设备的简单逻辑控制到运动控制、过程控制及集散控制等各种复杂任务的跨越。现在的 PLC 在模拟量处理、数学运算、人机接口和工业控制网络等各个方面的能力都已大幅提高，成为了工业控制领域的主流控制设备之一。

围绕 PLC 的应用，现场总线控制系统因其开放性、分散性以及完全可互操作性等特点，已成为未来新型工控领域的发展方向。现场总线技术是一种新型的工业控制技术，发展于 20 世纪 80 年代中期，用于工业生产现场，是一种在现场设备之间、现场设备与控制装置之间实现双向、互连、串行、多节点的数字通信技术。90 年代现场总线控制技术引入中国，结合 Intranet 和 Internet 的迅猛发展，日益显示出其传统控制系统无可替代的优越性，成为各大开发公司的争先研究热点之一。

现场总线技术的发展，进一步推动了工业控制系统数据交换与系统集成，但由于多种原因，目前多种现场总线协议标准共存是不争的客观事实，随着现场总线系统在工业企业中的广泛应用，对多种不同协议标准的现场总线之间实现相互信息交换已成为工业企业急需解决的问题。因此现场总线控制技术、系统集成技术及其应用的研究也一直是学术界和工程界研究的热点，研究解决不同现场总线系统及现场总线系统与其他系统之间的集成具有重要的实际意义。

随着工业生产规模的不断扩大，过程控制复杂程度大大提高，企业信息化建设需求明显，这使得工业控制向着综合自动化和信息化的方向发展。工业通信网络和系统集成技术，作为企业综合自动化和信息化的基础，是企业实现先进控制、过程优化和高效生产管理的技术保证，对工业自动化领域的发展起着举足轻重的作用。

工控网 2013 年发布的“2013 中国 PLC 市场品牌占有率分析”文章显示，SIEMENS 公司在小型 PLC 市场占据第一位，份额略超过 30%；而中型 PLC 占到市场份额的 65%，主导地位更是明显。不失一般性，本书围绕国内用量较大的 S7-200/S7-300 PLC，讲解其软件系统、硬件组成、基本指令及应用，以及现场总线和系统集成等知识。

本书在编写时考虑到涉及的知识点多、内容广等特点，采用结合生产实际、以案例带动知识点的方式开展学习，并注重解决实际问题及如何使用目前工业领域广泛使用的现场总线的能力培养。

本书内容选择合理丰富、结构清楚、图文并茂、面向应用，适合大中专院校电气自动化、生产过程自动化、网络技术、楼宇自动化等专业作为教学用书，也可作为工程人员的培训教材或相关科研人员、电气设计人员的参考书。

全书分为三篇共 14 章。第一篇介绍 S7-200 PLC 软件系统、硬件结构、基本指令、通信

协议及其应用；第二篇介绍 S7-300 PLC 软件系统、硬件结构、基本指令、通信协议及其应用；第三篇介绍系统集成技术及其应用，包括 S7-200/S7-300 PLC 的总线集成、不同的现场总线控制系统之间以及现场总线控制系统与信息网络之间的集成，案例涉及 Profibus、Profinet、WinCC 以及 OPC 等技术，它们对大中型 PLC 控制技术、现场总线控制技术及其系统集成技术的理论研究和工程应用具有一定的参考价值。

本书由无锡职业技术学院姚晓宁和郭琼老师担任主编，刘志刚老师担任副主编。无锡职业技术学院智能工程制造中心冯笠、秦爽和谢瑞等技术人员参与了本书的编写。在本书编写过程中，江苏锡恩智能工业科技股份有限公司吕中亮、陆路及姬世林三位工程师就本书内容的形成给出了许多宝贵的意见和建议，在此深表谢意。

本书在编写过程中参考了大量书籍、文献及手册资料，在此向各位相关作者深表感谢；同时由于作者水平有限，且控制技术及网络技术在不断地发展和完善，难免有以偏概全或不恰当之处，敬请读者批评指正。

编　者

目　　录

前言
第 1 章　PLC 概述 ………… 1
1.1　PLC 的产生与发展 ………… 1
1.2　PLC 的特点与应用领域 ………… 2
1.2.1　PLC 的特点 ………… 2
1.2.2　PLC 的应用领域 ………… 2
1.3　PLC 的分类与主要产品 ………… 3
1.3.1　PLC 的分类 ………… 3
1.3.2　PLC 的主要产品 ………… 3
1.4　PLC 的基本结构及工作原理 ………… 5
1.4.1　PLC 的基本结构 ………… 5
1.4.2　PLC 的基本工作原理 ………… 6
1.5　PLC 控制系统与继电器控制系统的比较 ………… 7
1.6　PLC 的编程语言 ………… 9

第一篇　S7-200 PLC 基础

第 2 章　S7-200 PLC 的硬件与编程资源 ………… 12
2.1　S7-200 PLC 的硬件 ………… 12
2.1.1　S7-200 PLC 的硬件结构 ………… 12
2.1.2　S7-200 PLC 的外部接线 ………… 16
2.2　S7-200 PLC 的编程元件 ………… 18
2.2.1　S7-200 PLC 的数据类型及寻址方式 ………… 18
2.2.2　S7-200 PLC 的编程资源 ………… 20
2.3　编程软件和程序结构 ………… 22
2.4　STEP 7-Micro/WIN 编程软件的使用 ………… 23
2.4.1　主界面的认识 ………… 23
2.4.2　程序编辑 ………… 24
2.4.3　程序的调试及监控 ………… 27
第 3 章　S7-200 PLC 的指令系统 ………… 29
3.1　位逻辑指令 ………… 29
3.1.1　标准触点指令 ………… 29
3.1.2　取反指令 ………… 29
3.1.3　边沿脉冲指令 ………… 30

3.1.4 置位/复位指令……30
3.1.5 立即指令……31
3.1.6 逻辑电路块的连接指令……31
3.1.7 堆栈操作指令……32
3.2 定时器指令……33
3.3 计数器指令……34
3.4 程序控制指令……35
3.5 数据处理指令……38
3.6 梯形图编程规则……44
3.7 S7-200 PLC 的基本编程环节……46
3.7.1 用标准触点指令实现一个脉冲输出电路……46
3.7.2 用一个按钮产生启动和停止信号电路……46
3.7.3 分频电路……46
3.7.4 单稳态电路……47
3.7.5 定时范围的扩展电路……47
3.7.6 闪烁电路……49
3.7.7 时序控制电路……49
第 4 章 S7-200 PLC 程序设计及其应用……51
4.1 程序设计方法……51
4.2 继电器控制电路移植法……51
4.2.1 两台电动机顺序控制的实现……52
4.2.2 星形−三角形降压起动控制的实现……54
4.3 经验设计法……57
4.3.1 电动机正反转控制的实现……57
4.3.2 花样喷泉控制功能的实现……61
4.4 顺序控制设计法……64
4.4.1 液压动力滑台运动过程的实现……64
4.4.2 三台电动机顺序起停控制功能的实现……68
4.5 综合实例……73
4.5.1 闪烁电路在监控系统中的应用……73
4.5.2 自动打铃程序设计与实现……74
4.5.3 台车的呼车控制系统设计与实现……77
第 5 章 S7-200 PLC 的通信……81
5.1 S7-200 PLC 通信概述……81
5.2 PPI 通信协议及其应用……81
5.2.1 PPI 通信协议……81
5.2.2 S7-200 PLC 之间的 PPI 通信实现……83
5.3 Modbus 通信协议及其应用……87
5.3.1 Modbus 通信协议介绍……87

5.3.2 Modbus RTU 通信 …… 88
5.3.3 Modbus 协议的安装 …… 90
5.3.4 Modbus 地址 …… 90
5.3.5 S7-200 PLC 之间的 Modbus 通信实现 …… 92
5.4 USS 通信协议及其应用 …… 97
5.4.1 USS 通信协议及其相关指令 …… 97
5.4.2 基于 HMI 的 S7-200 PLC USS 通信功能实现 …… 100

第二篇 S7-300 PLC 基础

第 6 章 S7-300 PLC 的硬件与编程资源 …… 107
6.1 S7-300 PLC 系统简介 …… 107
6.2 S7-300 PLC 系统组成 …… 107
6.2.1 主要模块介绍 …… 108
6.2.2 CPU 的操作模式 …… 112
6.3 STEP 7 编程软件 …… 114
6.3.1 STEP 7 的安装 …… 114
6.3.2 STEP 7 的硬件接口 …… 117
6.3.3 STEP 7 的授权 …… 119
6.3.4 创建一个工程 …… 119
6.3.5 STEP 7 的程序块 …… 122
6.4 简单项目的建立及运行 …… 123
6.4.1 控制要求 …… 123
6.4.2 硬件组态 …… 123
6.4.3 程序编写 …… 125
6.4.4 系统联调 …… 126
第 7 章 S7-300 PLC 指令系统 …… 128
7.1 数据及寻址地址 …… 128
7.1.1 数据块（DB） …… 128
7.1.2 数据类型 …… 129
7.1.3 地址区域 …… 135
7.1.4 指令操作数 …… 136
7.1.5 CPU 中的寄存器 …… 137
7.1.6 寻址方式 …… 137
7.2 位逻辑指令 …… 141
7.2.1 触点指令 …… 141
7.2.2 置位和复位指令 …… 142
7.2.3 RS 和 SR 触发器 …… 142
7.2.4 跳变沿检测指令 …… 143
7.2.5 SET/CLR 指令 …… 144

7.3 定时器指令 …… 144
7.3.1 脉冲定时器（SP） …… 144
7.3.2 扩展脉冲定时器（SE） …… 145
7.3.3 接通延时定时器（SD） …… 146
7.3.4 保持型接通延时定时器（SS） …… 146
7.3.5 断开延时定时器（SF） …… 146
7.4 计数器指令 …… 147
7.4.1 加计数器（S_CU） …… 147
7.4.2 减计数器（S_CD） …… 148
7.4.3 加减计数器（S_CUD） …… 149
7.4.4 计数器的线圈指令 …… 150
7.5 CPU 时钟存储器 …… 150
7.6 数据处理指令 …… 152
7.6.1 传送指令 …… 152
7.6.2 比较指令 …… 152
7.6.3 数据转换指令 …… 153
7.7 数学运算指令 …… 155
7.8 控制指令 …… 159
7.8.1 逻辑控制指令 …… 159
7.8.2 程序控制指令 …… 160
7.9 逻辑块 …… 160
7.9.1 组织块及其应用 …… 160
7.9.2 FC 及其应用 …… 169
7.9.3 FB 及其应用 …… 176
第 8 章 S7-300 PLC 程序设计及应用 …… 181
8.1 程序结构 …… 181
8.2 具有多种工作方式的机械手 PLC 控制系统的设计 …… 182
8.3 多台设备报警控制系统的设计与实现 …… 192
8.3.1 控制要求 …… 192
8.3.2 硬件配置及系统资源的分配 …… 192
8.3.3 程序实现 …… 194
8.3.4 系统联调 …… 195
8.4 基于模拟量的液位控制系统设计与实现 …… 196
8.4.1 控制要求及硬件配置 …… 196
8.4.2 硬件电路设计 …… 196
8.4.3 电气原理接线图 …… 197
8.4.4 程序结构 …… 198
8.4.5 功能实现 …… 199
8.5 基于 PID 的位置控制系统设计与实现 …… 202

8.5.1 项目背景 …… 202
8.5.2 硬件选型 …… 202
8.5.3 电气原理接线图 …… 203
8.5.4 项目结构及硬件组态要点 …… 203
8.5.5 PID 功能 …… 206
8.5.6 触摸屏界面设计 …… 208
8.5.7 系统功能的实现 …… 218
第 9 章 S7-300 PLC 的通信 …… 220
9.1 MPI 通信协议及其应用 …… 220
9.1.1 MPI 通信协议 …… 220
9.1.2 全局数据包通信 …… 220
9.1.3 调用通信功能块通信 …… 224
9.2 PROFIBUS 通信协议及其应用 …… 227
9.2.1 PROFIBUS 通信协议 …… 227
9.2.2 PROFIBUS 的传输技术 …… 232
9.2.3 PROFIBUS 控制系统的配置 …… 235
9.2.4 GSD 文件 …… 237
9.2.5 S7-300 PLC 之间的 PROFIBUS-DP 通信 …… 239
9.2.6 S7-300 PLC 与 MM420 变频器的 PROFIBUS-DP 通信 …… 245
9.3 工业以太网协议及其应用 …… 250
9.3.1 工业以太网基础知识 …… 250
9.3.2 工业以太网的现状与发展前景 …… 253
9.3.3 S7-300 PLC 与 ET200S 的 PROFINET 通信 …… 255

第三篇 自动控制系统集成

第 10 章 控制系统集成及其方法 …… 264
10.1 系统集成的概念 …… 264
10.2 控制网络 …… 265
10.2.1 网络节点 …… 266
10.2.2 现场总线控制网络的任务 …… 266
10.3 总线控制系统集成 …… 267
10.3.1 现场总线控制系统集成框架 …… 267
10.3.2 现场总线控制系统集成的原则 …… 268
10.4 现场总线控制系统和网络的集成方法 …… 269
10.4.1 FCS 和 DCS 的集成方法 …… 269
10.4.2 FCS 和网络的集成方法 …… 271
10.4.3 FCS 和其他现场总线的集成方法 …… 273
第 11 章 S7-300 PLC 与 S7-200 PLC 的系统集成 …… 275
11.1 基于 PROFIBUS-DP 的系统构建与运行 …… 275

11.1.1 控制要求及硬件配置……275
11.1.2 从站的设置……275
11.1.3 主站的硬件组态……276
11.1.4 程序编写……280
11.1.5 系统联调……280
11.2 基于 PROFINET 的系统构建与运行……280
11.2.1 控制要求及硬件配置……280
11.2.2 S7-200 PLC 的通信配置……281
11.2.3 系统功能的实现……286
第 12 章 基于 WinCC 的异构网络系统集成……288
12.1 WinCC 软件介绍……288
12.2 WinCC 通信处理……289
12.3 S7-200 PLC 与 S7-300 PLC 的系统集成……289
12.3.1 控制要求及硬件配置……289
12.3.2 S7-200 PLC 与 WinCC 通信的建立……290
12.3.3 S7-300 PLC 与 WinCC 通信的建立……292
12.3.4 系统功能的实现……292
12.3.5 系统联调……294
12.4 三菱 Q 系列 PLC 与 S7-300 PLC 的系统集成……295
12.4.1 控制要求及硬件配置……295
12.4.2 GX Developer 软件介绍……296
12.4.3 Q PLC 以太网通信……298
12.4.4 PLC 与 WinCC 通信的建立……300
12.4.5 程序编写……301
12.4.6 系统联调……302
第 13 章 基于 OPC 技术的异构网络系统集成……303
13.1 KEPServer 软件介绍……303
13.2 SIMATIC NET 软件介绍……304
13.3 基于 KEPServer 的 FX_{2N} PLC 与 S7-300 PLC 的通信系统……305
13.3.1 控制要求及硬件配置……305
13.3.2 S7-300 PLC 与服务器的通信建立……305
13.3.3 通信变量设置……307
13.3.4 PLC 与 KEPServer 系统联调……308
13.3.5 WinCC 与 KEPServer 通信建立……309
13.3.6 PLC 之间的数据交换……310
13.4 基于 SIMATIC NET 的 OPC 服务器与 S7-300 PLC 的通信系统……311
13.4.1 控制要求及硬件配置……311
13.4.2 PC 站的配置与组态……311
13.4.3 S7-300 PLC 项目设计……317

13.4.4 数据通信测试……318
13.4.5 WinCC 与 OPC 服务器的连接……320
13.4.6 系统功能的实现……321
第 14 章 机加工自动生产线控制系统集成……322
14.1 系统方案设计……322
14.2 系统硬件配置及组态……323
14.3 RFID 信息识别功能的实现……325
14.4 打标系统的集成……327
14.5 MES 系统与 PCS 系统的集成……330
14.6 车间级物料小车控制系统集成……331
14.7 项目总结……334
参考文献……335

第 1 章　PLC 概述

1.1　PLC 的产生与发展

在工业生产过程中，传统的生产设备大多采用继电器-接触器控制系统（J-C 控制系统），它按照逻辑条件进行顺序动作，并按照逻辑关系进行联锁保护动作等。但由于传统的继电器-接触器控制系统存在着明显的缺点，如体积大、可靠性差、动作速度慢、接线复杂、功能单一、难以实现比较复杂的控制，因此其通用性和灵活性显得相对较差。

1968 年，美国最大的汽车制造商——通用汽车公司（GM 公司）为了适应生产工艺不断更新的需要，提出采用一种新型的工业控制器取代继电器-接触器控制装置的设想。该设想把计算机控制的优点（功能完备，灵活性、通用性好）和继电器-接触器控制的优点（简单易懂、使用方便、价格便宜）结合起来，将继电器-接触器控制的硬接线逻辑转变为计算机的软件逻辑编程，且要求编程简单、上手容易，使不熟悉计算机的人员也能很快掌握其使用技术。第二年，美国数字设备公司（DEC 公司）研制出了第一台可编程序控制器，并在美国通用汽车公司的自动装配线上试用成功，取得了满意的效果，可编程序控制器自此诞生。

早期的可编程序控制器称为 Programmable Logic Controller（可编程逻辑控制器），简称 PLC，主要用于替代传统的继电器-接触器控制系统。随着 PLC 技术的不断发展，其功能日益丰富；1980 年，美国电气制造商协会（NEMA）给它取了一个新的名称“Programmable Controller”，简称 PC。为了避免与个人计算机（Personal Computer，也简称为 PC）这一简写名称混淆，故仍沿用早期的名称 PLC 表示可编程序控制器，但并不意味 PLC 只具有逻辑功能。

可编程序控制器是以微处理器为基础，综合了计算机技术、自动控制技术和通信技术而发展起来的一种新型、通用的自动控制装置。它是“专为在工业环境下应用而设计”的计算机。这种工业计算机采用“面向用户的指令”，因此编程方便。它能完成逻辑运算、顺序控制、定时、计数和算术操作，还具有“数字量或模拟量的输入/输出控制”等能力。

PLC 的定义有许多种，国际电工委员会（IEC）对 PLC 的定义是：可编程序控制器是一种专为在工业环境下应用而设计的数字运算操作的电子装置。它采用可编程序的存储器，用来在其内部存储执行逻辑运算、顺序控制、定时、计数和算术运算等操作的指令，并通过数字的或模拟的输入和输出，控制各种类型的机械或生产过程。可编程序控制器及其有关的外围设备，都应按易于与工业控制系统形成一个整体，易于扩展其功能的原则而设计。

20 世纪 80 年代至 90 年代中期，是 PLC 发展最快的时期，年增长率一直保持在 30%～40%。这一时期，PLC 在处理模拟量能力、数字运算能力、人机接口能力和网络能力等方面得到大幅度提高，同时 PLC 逐渐进入过程控制领域，在某些方面逐步取代了在过程控制领域处于统治地位的集散控制系统（DCS）。目前，世界上有 200 多个厂家生产 300 多个品种的 PLC 产品，应用在汽车、粮食加工、化学、制药、金属、矿山和造纸等多个行业。

PLC 具有通用性强、使用方便、适应面广、可靠性高、抗干扰能力强和编程简单等特点，已成为当前工业自动化领域使用量最多的控制设备，并跃居工业自动化三大支柱（PLC、机器人和 CAD/CAM）的首位。

1.2 PLC 的特点与应用领域

1.2.1 PLC 的特点

1．抗干扰能力强、可靠性高

在工业现场存在着电磁干扰、电源波动、机械振动、温度和湿度的变化等因素，这些因素都会影响到计算机的正常工作。而 PLC 在硬件和软件两个方面都采取了一系列的抗干扰措施，使其能够安全可靠地工作在恶劣的工业环境中。

在硬件方面，PLC 采用大规模和超大规模的集成电路，以及隔离、滤波、屏蔽、接地等抗干扰措施，并采取了耐热、防潮、防尘、抗震等措施；软件上，PLC 采用周期扫描工作方式，减少了由于外界环境干扰而引起的故障；系统程序中设有故障检测和自诊断程序，能对系统硬件电路等故障实现检测和判断，并采用数字滤波等抗干扰和故障诊断措施。以上这些使得 PLC 具有较高的抗干扰能力和可靠性。

2．控制系统结构简单、使用方便

在 PLC 控制系统中，只需在 PLC 的输入/输出端子上接入相应的信号线即可，不需要连接时间继电器、中间继电器之类的电压电器和大量复杂的硬件接线，这大大简化了控制系统的结构。PLC 体积小、质量轻，安装与维护也极为方便。另外 PLC 的编程大多采用类似于继电器控制线路的梯形图形式，这种编程语言形象直观、容易掌握，编程非常方便。

3．功能强大、通用性好

PLC 内部有大量可供用户使用的编程元件，具有很强的功能，可以实现非常复杂的控制功能。另外 PLC 的产品已经标准化、系列化、模块化，配备有品种齐全的各种硬件装置供用户选用，用户能灵活方便地进行系统配置，以组成不同功能、不同规模的控制系统。

1.2.2 PLC 的应用领域

随着 PLC 技术的发展，PLC 的应用领域已经从最初的单机、逻辑控制发展到能够联网的、功能丰富的控制。目前，PLC 已广泛应用于钢铁、石油、化工、电力、建材、机械制造、汽车、轻纺、交通运输、环保及文化娱乐等各个行业。

1．逻辑控制

PLC 最初能完成的基本功能是通过“与”、“或”、“非”等逻辑指令的组合，代替继电器进行组合逻辑控制、定时控制与顺序逻辑控制。例如印刷机、注塑机、组合机床、电镀流水线和电梯控制等。

2．运动控制

PLC 可以使用专用的运动控制模块，对步进电动机或伺服电动机的单轴或多轴的位置进行控制。PLC 将描述位置的数据送给模块，其输出移动一轴或数轴到目标位置。每个轴移动时，位置控制模块保持适当的速度和加速度，以确保运动平滑。例如各种机械、机床、机器

人和电梯等场合。

3．过程控制

过程控制是指对温度、压力以及流量等模拟量的控制。对于温度、压力以及流量等模拟量，PLC 提供了配套的模数（A-D）和数模（D-A）转换模块，使 PLC 可以方便地处理这些模拟量；PLC 还提供了 PID 功能指令，可以方便地进行闭环控制，从而实现过程控制。过程控制在冶金、化工、热处理及锅炉控制等场合有着非常广泛的应用。

4．工业控制网络分级系统

PLC 能与计算机、PLC 及其他智能装置联成网络，使设备级的控制、生产线的控制及工厂管理层的控制构成一个整体，形成控制自动化与管理自动化的有机集成，从而创造更高的企业效益。

1.3 PLC 的分类与主要产品

1.3.1 PLC 的分类

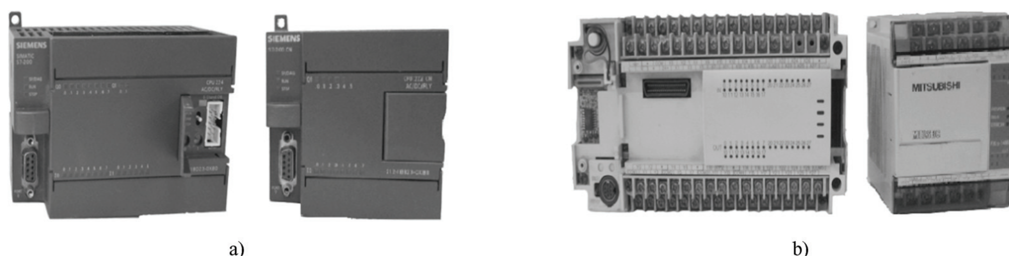
PLC 的分类有以下两种方法。

1．按 PLC 的点数来分类

根据 PLC 可扩展的输入/输出点数，可以将 PLC 分为小型、中型和大型三类。小型 PLC 的输入/输出点数在 256 点以下；中型 PLC 的输入/输出点数为 256～2048 个点；大型 PLC 的输入/输出点数在 2048 点以上。

2．按 PLC 的结构分类

按 PLC 的结构分类，PLC 可分为整体式和模块式。整体式 PLC 将电源、CPU、存储器及 I/O 系统都集中在一个小箱体内，小型 PLC 多为整体式 PLC，如图 1-1 所示；模块式 PLC 是按功能分成若干模块，如电源模块、CPU 模块、输入模块、输出模块以及通信模块等，再根据系统要求，组合不同的模块，形成不同用途的 PLC，大中型的 PLC 多为模块式 PLC，如图 1-2 所示。



a)　　　　b)

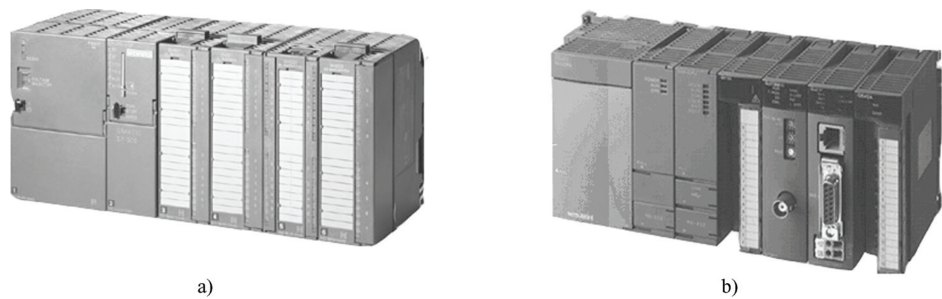
图 1-1　整体式 PLC

a) 西门子 S7-200 系列　b) 三菱 FX_{2N}/FX_{1S} 系列

1.3.2 PLC 的主要产品

目前全球 PLC 生产厂家有 200 多家，比较著名的有德国的西门子（SIEMENS），美国的 AB、通用（GE）、莫迪康（MODICON），日本的三菱（MITSBISHI）、欧姆龙（OMRON）、

富士电机（FUJI）、松下电工，法国的 TE（TELEMECANIQUE）、施耐德（SCHNEIDER），韩国的三星（SUMSUNG）与 LG 等。



a) b)

图 1-2 模块式 PLC

a) 西门子 S7-300 系列 b) 三菱 Q 系列

德国西门子公司的 PLC 产品在国际上具有较高的市场占有率，其主要产品有 S5、S7、C7、M7 及 WinAC 等几个系列。其中 S7 系列 PLC 是目前市场的主流产品，分为 LOGO!、S7-200、S7-1200、S7-300、S7-400 几个子系列。

其中 S7-200 系列 PLC 采用整体式、紧凑型结构，是针对简单控制系统使用的小型 PLC，一般适用于 I/O 点数为 100 点左右的单机设备或小型控制系统。

S7-1200 系列 PLC 采用模块化、紧凑型结构，是针对具有人机界面的简单控制系统而设计的小型 PLC，一般适用于要求简单、具有 HMI 和网络功能的小型自动化系统。

S7-300 系列 PLC 采用模块化、无风扇结构，是针对中小型控制系统设计的中型 PLC，一般适用于 I/O 点数为 1000 点左右的集中或分布式的中小型自动化控制系统。

S7-400 系列 PLC 采用模块化、无风扇结构，是针对大中型控制系统设计的大型 PLC，一般适用于 I/O 点数为 10000 点左右的大型自动化控制系统。

我国的 PLC 研制、生产和应用也发展很快。在 20 世纪 70 年代末至 80 年代初，我国引进了不少国外的 PLC 成套设备。此后，在传统设备改造和新设备设计中，PLC 的应用逐年增多，并取得显著的经济效益。我国从 90 年代开始生产 PLC，也拥有较多的 PLC 品牌，如台湾地区的永宏、台达、盟立等，北京的和利时、凯迪恩（KDN）等，浙大中控、中自、无锡信捷、深圳汇川等；目前应用较广的 PLC 生产厂家的主要产品见表 1-1。

表 1-1 部分 PLC 生产厂家及主要产品

国　家	公　司	产 品 型 号
德国	西门子 SIEMENS	LOGO!、S7-200、S7-1200、S7-300、S7-400 系列
美国	GE Fanuc	90^{TM}-30 系列、90^{TM}-70 系列、VersaMax 系列
日本	三菱 MITSUBISHI	FX_{1S}、FX_{2N}、FX_{3U} 系列，A 系列，Q 系列
法国	施耐德 SCHNEIDER	Twido、Micro、Premume、Compaq 系列
中国	无锡信捷	XC1、XC2、XC3、XC5 系列
中国	深圳汇川	H0U、H1U、H2U、H3U 系列

1.4 PLC 的基本结构及工作原理

1.4.1 PLC 的基本结构

各种 PLC 的组成结构基本相同。主要由 CPU、电源、存储器和输入/输出接口电路等组成。小型 PLC 将 CPU、电源和数字量输入/输出接口等集成为一个整体模块，其结构如图 1-3 所示。

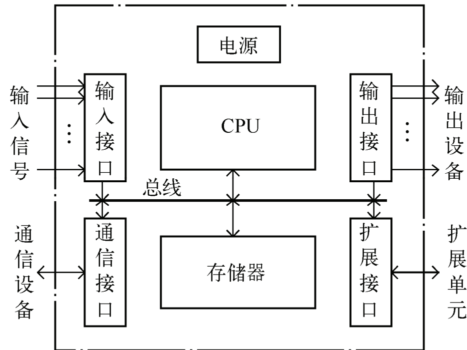


图 1-3　PLC 的基本结构

1．中央处理单元（CPU）

中央处理器单元（CPU）是 PLC 的核心部件，一般由控制器、运算器和寄存器组成。CPU 通过地址总线、数据总线、控制总线与存储单元、输入/输出接口、通信接口以及扩展接口相连。它不断地采集输入信号，执行用户程序，刷新系统的输出。

2．存储器

PLC 的存储器包括系统存储器和用户存储器两种。系统存储器用于存放 PLC 厂家编写的系统程序，以及开机自检、程序解释等功能，用户不能访问和修改，一般固化在只读存储器 ROM 中；用户存储器用于存放 PLC 的用户程序，设计和调试时需要不断修改，一般存放在读写存储器 RAM 中；当用户调试好的程序需要长期使用时，也可选择将其写入可电擦除的 E^2PROM 存储卡中，实现长期保存。

3．输入/输出接口（I/O）单元

PLC 的输入/输出接口单元是 CPU 与外部设备连接的桥梁，通过 I/O 接口，PLC 可实现对工业设备或生产过程的参数检测和过程控制。输入接口电路的作用是将按钮、行程开关或传感器等产生的信号送入 CPU；PLC 的输出接口电路的作用是将 CPU 向外输出的信号转换成可以驱动外部执行元件的信号，以便控制接触器线圈、电磁阀和指示灯等外部电器的通、断电。PLC 的输入/输出接口电路一般采用光电耦合隔离技术，可以有效地保护内部电路。

1）输入接口电路

PLC 的输入接口电路可分为直流输入电路和交流输入电路。直流输入电路的延迟时间比较短，可以直接与接近开关、光电开关等电子输入装置连接；交流输入电路适用于在油雾、粉尘等恶劣环境下使用。

直流输入电路如图 1-4 所示，图中只画出了一路直流输入电路，方框内为 PLC 输入内部电路，方框外为外部信号输入电路。当外部开关接通时，输入信号为“1”，直流 24V 经限流电阻、RC 滤波电路和光电耦合电路后送给 PLC 内部。

交流输入电路与直流输入电路类似，外接的输入电源为 220V 交流电源。

2）输出接口电路

输出接口电路通常有 3 种类型：继电器输出型、晶体管输出型和晶闸管输出型。

继电器输出的优点是电压范围宽，导通压降小，价格便宜，既可以控制直流负载，也可以控制交流负载；缺点是触点寿命短，转换频率低。

晶体管输出的优点是寿命长、无噪声、可靠性高，转换频率快，可驱动直流负载；缺点是过载能力较差。

晶闸管输出的优点是寿命长、无噪声、可靠性高，可驱动交流负载；缺点是过载能力较差。

继电器输出电路如图 1-5 所示，图中只画出了一路继电器输出电路，方框内为 PLC 输出接口内部电路，方框外为外部负载控制电路。当输出为“1”时，接通内部继电器线圈，触点闭合，从而控制外部电路导通，负载开始动作。

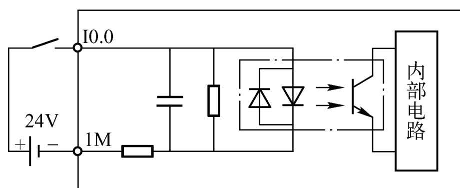


图 1-4　直流输入电路

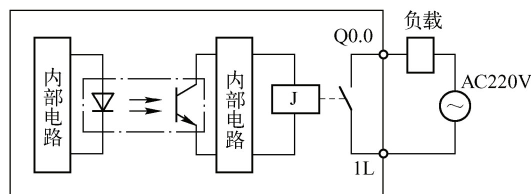


图 1-5　继电器输出电路

晶体管输出型、晶闸管输出型和继电器输出型的输出电路类似，只是用晶体管或晶闸管代替继电器来控制外部负载。

4．扩展接口和通信接口

PLC 的扩展接口的作用是将扩展模块和功能模块与基本单元相连，使 PLC 的配置更加灵活，控制功能更为丰富，以满足不同控制系统的需要；通信接口的功能是通过这些通信接口可以和监视器、打印机以及其他的 PLC 或者计算机相连，从而实现“人-机”或“机-机”之间的对话。

5．电源

PLC 一般使用 220V 交流电源或 24V 直流电源，内部的开关电源为 PLC 的中央处理器、存储器等电路提供 5V、12V 和 24V 直流电源，使 PLC 能正常工作。

1.4.2　PLC 的基本工作原理

PLC 有两种工作方式，即 RUN（运行）方式和 STOP（停止）方式。在 RUN 方式中，CPU 才执行用户程序，并输出运算结果；在 STOP 方式中，CPU 不执行用户程序，但可将用户程序和硬件设置信息下载到 PLC 中。

PLC 控制系统与继电器控制系统在运行方式上存在着本质的区别。继电器控制系统采用的是“并行运行”的方式，各条支路同时上电，当一个继电器的线圈通电或者断电，该继电器的所有触点都会立即同时动作；而 PLC 采用“周期循环扫描”工作方式，CPU 是通过逐行扫描并执行用户程序来实现的，即如果一个逻辑线圈接通或断开，该线圈的所有触点并不会

立即动作，必须等到扫描执行到该触点时才会动作。

一般来说，当 PLC 运行后，其工作过程可分为输入采样阶段、程序执行阶段和输出刷新阶段。完成上述 3 个阶段即称为完成一个扫描周期。

在整个运行期间，PLC 的 CPU 以一定的扫描速度重复执行上述 3 个阶段。PLC 的扫描工作过程如图 1-6 所示。图中，输入映像寄存器是指在 PLC 的存储器中设置一块用来存放输入信号的存储区域，而输出映像寄存器是用来存放输出信号的存储区域；元件映像存储器是包括输入和输出映像寄存器在内的所有 PLC 梯形图中编程元件的映像存储区域的统称。

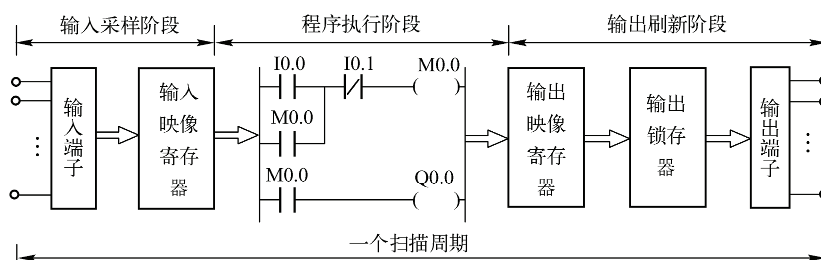


图 1-6　PLC 的扫描工作过程

1．输入采样阶段

PLC 将各输入状态存入对应的输入映像寄存器中，此时，输入映像寄存器被刷新，接着进入程序执行阶段。在程序执行阶段或输出刷新阶段，输入元件映像寄存器与外界隔绝，无论输入端子信号怎么变化，其内容保持不变，直到下一个扫描周期的输入采样阶段才将输入端子的新内容重新写入。

2．程序执行阶段

PLC 根据最新读入的输入信号，以从左到右、从上到下的顺序逐行扫描，执行一次程序，结果存入输出元件映像寄存器中。对于元件映像寄存器，每个元件（除输入映像寄存器外）的状态会随着程序的执行而变化。

3．输出刷新阶段

在所有指令执行完毕后，输出映像寄存器中所有输出继电器的状态（“1”或“0”）在输出刷新阶段统一转存到输出锁存器中，并通过一定的方式输出驱动外部负载。

1.5　PLC 控制系统与继电器控制系统的比较

继电器-接触器控制是采用硬件和接线来实现的，它通过选用合适的分立元件（接触器、主令电器、各类继电器等），然后按照控制要求用导线将触点相互连接，从而形成并实现既定的逻辑控制；如控制要求改变，硬件构成及接线都需相应调整。

而 PLC 控制采用程序存储器控制，其控制逻辑是以程序方式存储在内存中，系统要完成的控制任务是通过执行存放在存储器中的程序来实现的；如控制要求改变，硬件电路连接可不用调整或简单改动，主要通过改变程序即可，也称为“软接线”。

下面以电动机星-三角降压起动控制为例，分别采用继电器-接触器控制和 PLC 控制方式实现，分析和总结这两种控制方式的异同点。

继电器-接触器控制方式如图 1−7 所示，主电路和控制电路由导线相互连接分立元件的各端子构成，其控制逻辑包含于控制电路中，通过连线体现。

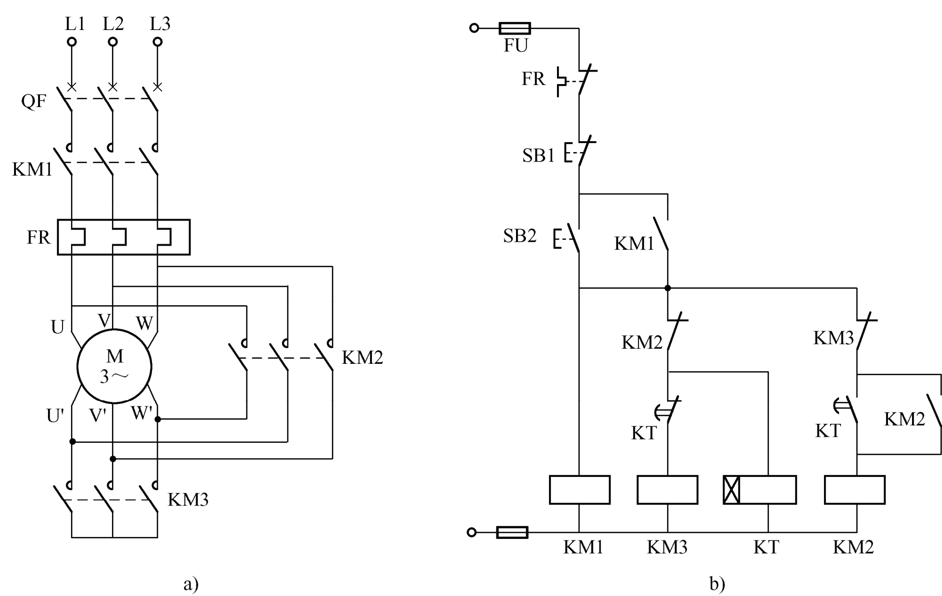


图 1−7　星-三角降压起动继电接触器控制方式

a) 主电路　b) 控制电路

PLC 控制方式如图 1−8 所示，其主电路不变，控制电路由 PLC 接线图和程序两部分实现；而控制逻辑是通过软件即编制相应程序实现的。

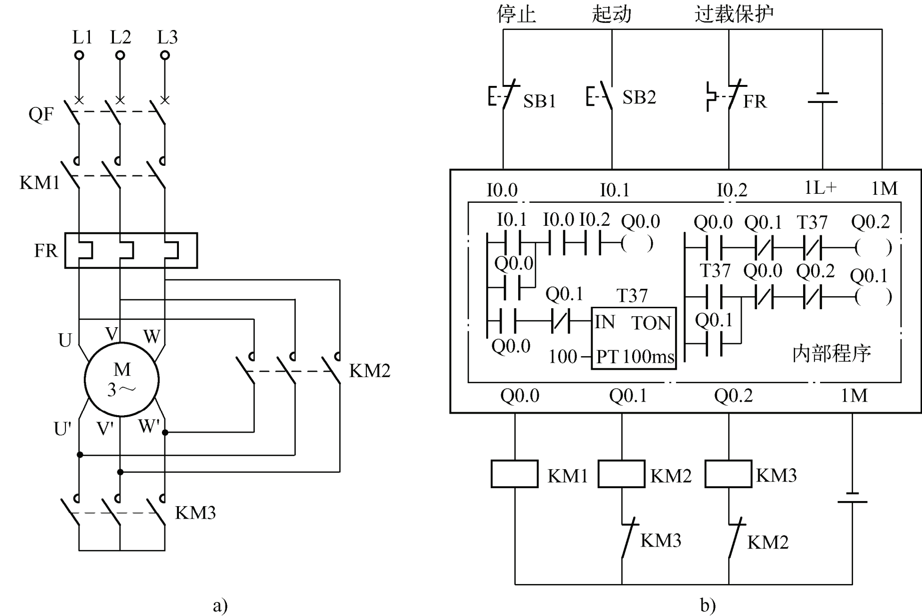


图 1−8　星-三角降压起动 PLC 控制方式

a) 主电路　b) 控制电路

比较 PLC 控制与继电器-接触器控制两种控制方式：

1）PLC 控制系统与继电器-接触器控制系统的输入、输出部分基本相同，输入部分都是由按钮、开关、传感器等组成；输出部分都是由接触器、执行器、电磁阀等部件构成。

2）PLC 控制采用软件编程取代了继电器-接触器控制系统中大量的中间继电器、时间继电器和计数器等器件，使控制系统的体积、安装及接线工作量大大减少；可以有效减少系统维修工作量，并提高工作可靠性。

3）PLC 控制系统不仅可以替代继电器-接触器控制系统，而且当生产工艺、控制要求发生简单调整时，不需要重新连线，只需相应修改程序或配合程序对硬件接线做很少的变动就可以了。

4）PLC 控制系统除了可以完成传统继电器-接触器控制系统所具有的功能外，还可以有效地拓展控制功能，实现模拟量控制、高速计数、开环或闭环过程控制以及通信联网等功能。

1.6 PLC 的编程语言

PLC 程序是设计人员根据控制系统的实际控制要求，通过 PLC 的编程语言进行编制的。根据国际电工委员会制定的工业控制编程语言标准（IEC1131-3），PLC 的编程语言有以下五种，分别为梯形图（Ladder Diagram，LD）、语句表（Instruction List，IL）、顺序功能图（Sequential Function Chart，SFC）、功能块图（Function Block Diagram，FBD）及结构化文本（Structured Text，ST）。

1．梯形图语言（LD）

梯形图语言是 PLC 程序设计中最常用的编程语言，它是与继电器线路类似的一种图形化的编程语言。由于电气设计人员对继电器控制较为熟悉，因此，梯形图编程语言得到了广泛的欢迎和应用。

梯形图编程语言的特点是：与电气操作原理图相对应，具有直观性和对应性；与原有继电器控制相一致，使电气设计人员易于掌握。

梯形图编程语言与原有的继电器控制的不同点是：梯形图中的能流不是实际意义的电流，内部的继电器也不是实际存在的继电器，应用时需要与原有继电器控制的概念区别对待。

图 1-9 是典型电动机星-三角起动控制电路图和对应的采用 PLC 控制的程序梯形图。

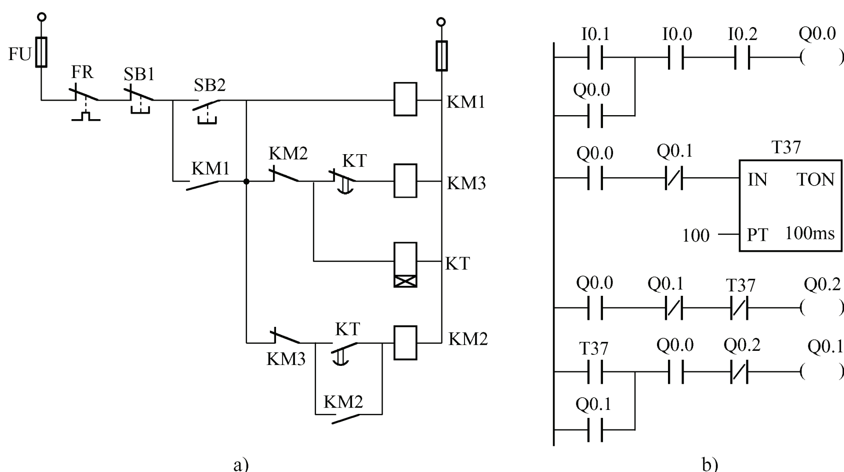


图 1-9　电动机星-三角起动控制电路和对应梯形图

a) 电动机星-三角起动控制电路　b) 星-三角起动对应梯形图

2．指令表语言（IL）

指令表编程语言是与汇编语言类似的一种助记符编程语言，和汇编语言一样由操作码和操作数组成。在无计算机的情况下，适合采用 PLC 手持编程器对用户程序进行编制。同时，指令表编程语言与梯形图编程语言图一一对应，在 PLC 编程软件下可以相互转换。

图 1-10 为一段 PLC 梯形图及其对应的指令表。

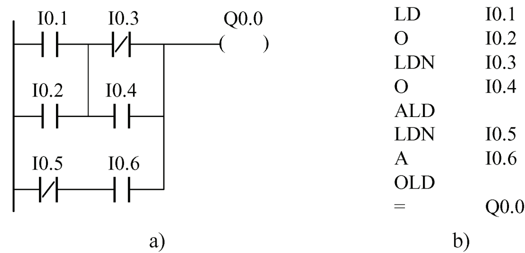


图 1-10　PLC 梯形图和其对应的指令表

a) 梯形图　b) 对应指令表

3．顺序功能图语言（SFC）

顺序功能图语言是为了满足顺序逻辑控制而设计的编程语言。编程时将顺序流程动作的过程分成步和转换条件，根据转移条件对控制系统的功能流程顺序进行分配，一步一步的按照顺序动作。每一步代表一个控制任务，用方框表示。在方框内含有用于完成相应控制功能任务的梯形图逻辑。这种编程语言使程序结构清晰，易于阅读及维护，大大减轻编程的工作量，缩短编程和调试时间。多适用于系统的规模较大且程序关系较复杂的场合。

图 1-11 是一个简单的顺序功能流程编程语言的示意图。

4．功能模块图语言（FBD）

功能模块图语言是与数字逻辑电路类似的一种 PLC 编程语言。采用功能模块图的形式来表示模块所具有的功能，不同的功能模块有不同的功能。基本沿用了半导体逻辑电路的逻辑方块图，有数字电路基础的技术人员很容易掌握。

图 1-12 是某一程序的功能模块图语言的示意图。

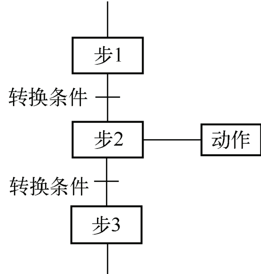


图 1-11　顺序功能流程编程语言示意图

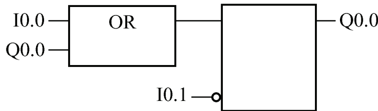


图 1-12　功能模块图

5．结构化文本语言（ST）

结构化文本语言是用结构化的描述文本来描述程序的一种编程语言。它是类似于高级语言的一种编程语言。在大中型的 PLC 系统中，常采用结构化文本来描述控制系统中各个变量的关系。主要用于其他编程语言较难实现的用户程序编制。

结构化文本编程语言采用计算机的描述方式来描述系统中各种变量之间的各种运算关系，完成所需的功能或操作。大多数 PLC 制造商采用的结构化文本编程语言与 BASIC 语言、PASCAL 语言或 C 语言等高级语言相类似，但为了应用方便，在语句的表达方法及语句的种类等方面都进行了简化。

结构化文本编程语言的特点是：采用高级语言进行编程，可以完成较复杂的控制运算；需要有一定的计算机高级语言的知识和编程技巧，对工程设计人员要求较高；直观性和操作性较差。

不同型号的 PLC 编程软件对以上五种编程语言的支持种类是不同的，早期的 PLC 仅仅支持梯形图编程语言和指令表编程语言。目前的 PLC 对梯形图（LD）、指令表（STL）、功能模块图（FBD）编程语言都予支持，例如 SIMATIC STEP 7 MicroWIN V3.2；而在大中型 PLC 编程中，ST 语言应用越来越广，可以非常方便地描述控制系统中各个变量的关系。在 PLC 控制系统设计中，要求设计人员不但要对 PLC 的硬件性能了解外，也要了解 PLC 对编程语言支持的种类。

第一篇　S7-200 PLC 基础

第 2 章　S7-200 PLC 的硬件与编程资源

2.1　S7-200 PLC 的硬件

2.1.1　S7-200 PLC 的硬件结构

德国西门子（SIEMENS）公司的 SIMATIC 系列PLC经历了 C3、S3、S5 及 S7 系列，已成为应用非常广泛的可编程控制器。产品发展至今，S3、S5 系列 PLC 已逐步退出市场，而 S7 系列 PLC 已发展成为西门子自动化系统的控制核心。

西门子 S7-200 系列 PLC 是 SIMENS 公司生产的小型 PLC，属于西门子 S7-200/300/400 PLC 家族中功能最精简、I/O 点数最少、扩展性能最低的产品。S7-200 系列 PLC 的基本单元将 CPU、电源、I/O 点集成在一个封装中，其结构紧凑，功能强大，使用范围可覆盖从替代继电器的简单控制到更复杂的自动化控制。主要应用于与自动检测、自动化控制有关的工业及民用领域，包括各种机床、机械、电力设施、民用设施及环保设备等。

S7-200 系列 PLC 构建的控制系统一般由一个必备的基本单元和若干个可选择的扩展模块组成，用户可根据系统的实际控制要求进行灵活配置。

1．基本单元

S7-200 系列包括 CPU221、CPU222、CPU224、CPU224XP 和 CPU226 共 5 种型号的基本单元，其主要技术指标见表 2-1，外形结构如图 2-1 所示。

表 2-1　S7-200 CPU 主要技术指标

特性	CPU221	CPU222	CPU224	CPU224XP	CPU226
数据存储器	2048	2048	8192	10240	10240
本机数字量 I/O	6 入/4 出	8 入/6 出	14 入/10 出	14 入/10 出	24 入/16 出
本机模拟量 I/O	—	—	—	2 入/1 出	—
扩展模块数量	—	2	7	7	7
高速计数器个数	4	4	6	6	6
单相高速计数器个数	4 路 30kHz	4 路 30kHz	6 路 30kHz	4 路 30kHz 或 2 路 200kHz	4 路 30kHz
双相高速计数器个数	2 路 20kHz	2 路 20kHz	4 路 20kHz	3 路 20kHz 或 1 路 100kHz	2 路 20kHz
高速脉冲输出	2 路 20kHz	2 路 20kHz	2 路 20kHz	2 路 100kHz	2 路 20kHz
RS-485 通信口	1 个	1 个	1 个	2 个	2 个
支持的通信协议	PPI/MPI/自由口	PPI/MPI/自由口/PROFIBUS-DP			

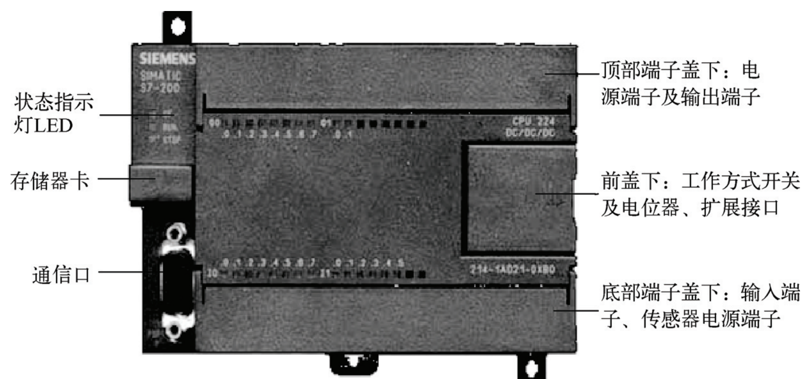


图 2-1　CPU224 DC/DC/DC 外形图

CPU221 为整体式固定 I/O 结构，无扩展功能，属于微型控制器。其余 4 种（CPU222、CPU224、CPU224XP 和 CPU226）均为基本单元加扩展的结构，其中 CPU224XP 是 S7-200 系列的升级产品，它另外集成有 2 路模拟量输入（10 位，±DC10V），1 路模拟量输出（10 位，DC0~10V 或 DC4~20mA），有 2 个 RS-485 通信口，高速脉冲输出频率提高到 100kHz，双相高速计数器频率提高到 100kHz。CPU226 适用于复杂的中小型控制系统，可扩展到 248 点数字量和 35 路模拟量，有 2 个 RS-485 通信接口。

S7-200 PLC 的 CPU 模块均集成有一定数量的输入点，输入点内部带有双向光耦输入元件，其数字量输入技术指标见表 2-2。

表 2-2　S7-200 数字量输入技术指标

项　　目	DC24V 输入（除 CPU224XP 外）	DC24 输入（CPU224XP）
输入类型	漏型/源型（IEC 类型 1）	漏型/源型（IEC 类型 1，I0.3~I0.5 除外）
输入电压额定值	DC24V，典型值 4mA	
输入电压浪涌值	35V/0.5s	
逻辑 1 信号	≥DC15V/2.5mA	I0.3~I0.5 为 DC4V/8mA；其余为 DC15V/2.5mA
逻辑 0 信号	≤DC5V/1.0mA	I0.3~I0.5 为 DC1V/1mA；其余为 DC5V/1mA
输入延迟	0.2~12.8ms（可选择）	
连接 2 线式接近开关的允许漏电流	最大 1.0mA	
光电隔离	AC500V，1min	
高速计数器输入	逻辑 1 电平 DC15~30V：单相 20kHz，两相 10kHz；逻辑 1 电平 DC15~26V：单相 30kHz，两相 20kHz	
CPU224XP 的 HSC4 和 HSC5	逻辑 1 电平 > DC4V 时，单相 200kHz，两相 100kHz	
电缆长度	非屏蔽 300m，屏蔽电缆 500m，HSC 50m	

S7-200 PLC 的 CPU 模块均集成有一定数量的输出点，当采用 CPU 模块为 DC 电源输入时，输出采用直流晶体管驱动；当采用 CPU 模块为 AC 电源输入时，输出采用继电器接点驱动；输出均带有公共端，但点数不一。输出点内部带有光耦元件，其 CPU 模块的数字量输出技术指标见表 2-3。

表 2-3　S7-200 数字量输出技术指标

输出类型	晶体管输出（除 CPU224XP 外）	晶体管输出（CPU224XP）	继电器输出
输出电压额定值	DC24V	DC24V	DC24V 或 AC250V
输出电压允许范围	DC20.4~28.8V	DC5~28.8V（Q0.0~Q0.4）	DC5~30V
		DC20.4~28.8V（Q0.5~Q1.1）	AC5~250V
浪涌电流	最大 8A，100ms		5A，4s，占空比 0.1
逻辑 1 信号	≥DC20V/0.75A		—
逻辑 0 信号	≤DC0.1V/10kΩ负载		—
逻辑 1 最大输出电流	≤0.75A/点；公共端≤6A	≤0.75A/点；公共端≤3.75A	≤2A/点；公共端≤10A
逻辑 0 最大漏电流	10μA	10μA	—
白炽灯负载	5W	5W	直流负载 30W/交流负载 200W
接通状态电阻	0.3Ω，最大 0.6Ω	0.3Ω，最大 0.6Ω	新的时最大 0.2Ω
输出响应时间（接通）	Q0.0 和 Q0.1 为 2μs，其他 15μs	Q0.0 和 Q0.1 为 0.5μs，其他 15μs	≈10ms
输出响应时间（断开）	Q0.0 和 Q0.1 为 10μs，其他 130μs	Q0.0 和 Q0.1 为 1.5μs，其他 130μs	≈10ms
最高脉冲频率	20kHz（Q0.0 和 Q0.1）	100kHz（Q0.0 和 Q0.1）	1Hz
触点机械寿命	—	—	10000000 次，无负载
额定负载时触点寿命	—	—	100000 次，额定负载
电缆长度	非屏蔽 300m，屏蔽电缆 500m，HSC 50m		

2．扩展模块

为了更好地满足实际工程应用要求，S7-200 系列 PLC 提供了多种类型的扩展模块，可以通过选用不同功能的扩展模块与基本单元结合，进一步完善和发挥 PLC 的功能。但扩展模块只能与基本单元配合使用，不能单独构成系统。

S7-200 系列 PLC 的基本单元与各类扩展模块之间、扩展模块与扩展模块之间均通过扩展模块自带的电缆进行连接。

（1）数字量 I/O 扩展模块

S7-200 系列 PLC 提供了多种数字量 I/O 扩展模块，包括 EM221、EM222、EM223 扩展模块。可提供 8 点、16 点或 32 点数字量 I/O 点，以满足不同的控制需要。除 CPU221 外，其他 CPU 模块均可以配接多个扩展模块，连接时 CPU 模块放在最左侧，扩展模块用扁平电缆与左侧的模块相连。具体数字量输入/输出模块见表 2-4，外形结构如图 2-2 所示。

表 2-4　数字量扩展模块

型　　号	各组输入点数	各组输出点数
EM221 8 点，DC24V 输入	4，4	—
EM221 8 点，AC230V 输入	8 点相互独立	—
EM221 16 点，DC24V 输入	4，4，4，4	—
EM222 4 点，DC24V/5A 输出	—	4 点相互独立
EM222 4 点，10A 继电器输出	—	4 点相互独立
EM222 8 点，DC24V 输出	—	4，4

（续）

型　　号	各组输入点数	各组输出点数
EM222 8 点，继电器输出	—	4，4
EM222 8 点，AC230V 输出	—	8 点相互独立
EM223 4 输入/4 输出，DC24V	4	4
EM223 DC24V 4 输入/继电器 4 输出	4	4
EM223 DC24V 8 输入/继电器 8 输出	4，4	4，4
EM223 8 输入/8 输出，DC24V	4，4	4，4
EM223 16 输入/16 输出，DC24V	8，8	4，4，8
EM223 DC24V 16 输入/继电器 16 输出	8，8	4，4，4，4



图 2-2　EM222 8 点继电器输出外形图

（2）模拟量扩展模块

S7-200 PLC 有 3 种模拟量扩展模块，见表 2-5，转换位数都为 12 位。外形结构如图 2-3 所示。

表 2-5　模拟量扩展模块

模块	EM231	EM232	EM235
点数	4 路模拟量输入	2 路模拟量输出	4 路模拟量输入，1 路模拟量输出

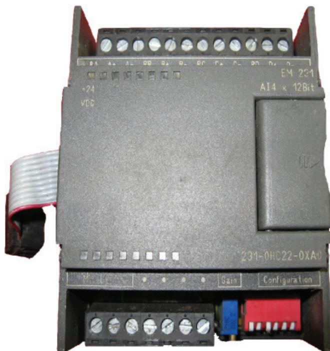


图 2-3　EM231 热电阻模块外形图

（3）热电偶/热电阻模块

EM231 热电偶/热电阻模块是最常用的模块，具有冷端补偿电路。EM231 热电偶输出的电压范围为±80mV，模块输出 15 位加符号位的二进制数。EM231 热电偶可以用于 J、K、E、N、S、T 和 R 型热电偶，用模块下方的 DIP 开关来选择热电偶的类型。EM231 热电阻的接线方式有 2 线、3 线和 4 线三种，4 线方式的精度最高，2 线方式受接线误差的影响精度最低；EM231 热电阻模块可以通过 DIP 开关来选择热电阻的类型、接线方式、测量单位和开路故障的方向。

3．特殊功能扩展模块

S7-200PLC 还提供了一些特殊功能模块，用于完成特定的任务。

1）EM253：该模块为定位控制模块，能提供 12Hz~200kHz 的脉冲频率，用于步进电动机和伺服电动机的速度和位置的开环控制。

2）EM277：PROFIBUS-DP 从站通信模块。在 PROFIBUS-DP 网络中，S7-200 PLC 与 EM277 模块配合，只能作从站。

3）EM241：Modem 通信模块。通过该模块可以实现 S7-200 PLC 通过电话交换机和电话网络的远距离通信。

4）CP243-1：工业以太网通信模块。通过该模块可以将 S7-200 PLC 连接在以太网上，使用时需要对其进行组态，支持 STEP 7-Micro/WIN32 软件。

5）CP243-1 IT：工业以太网通信模块，可提供 Web/E-mail 等 IT 应用。

6）CP243-2：AS-i 主站模块，与 S7-200 PLC 配合使用，使 S7-200 PLC 接入 AS-i 网络。

4．相关设备

S7-200 系列 PLC 提供了文本显示器 TD-200 和 TD-200C，它们可以显示两行，每行 20 个字符，每两个字符的位置可以显示 1 个汉字，通过它们可以查看、监控和改变应用程序的过程变量。S7-200 系列 PLC 支持 TP070 和 TP170 触摸屏，它们用专用的组态软件 PROTOOL 来生成画面，由用户自定义操作接口，例如图形、滚动条、按钮、指示灯以及输入框等。

2.1.2　S7-200 PLC 的外部接线

1．端子排

西门子 S7-200 PLC 的基本模块根据供电电源的类型、数字量输入/输出接点的类型，提供两种电源供电的 CPU 类型。其中 DC/DC/DC 表示该 CPU 是直流供电，直流数字量输入，晶体管输出；AC/DC/RELAY 表示该 CPU 是交流供电，直流数字量输入，继电器输出。

如图 2-4 所示，以 CPU224 DC/DC/DC 型号的 PLC 为例。L+、M 端是电源的输入端，使用 24V 直流电源，L+端接直流电源正极，M 端接直流电源负极。机内自带直流 24V 内部电源，为输入器件和扩展模块供电（图中的 DC24V 传感器电源输出）。

图 2-4 下侧 0.0~0.7 和 1.0~1.5 为输入端子，1M 为输入端子 0.0~0.7 的公共端子，2M 为输入端子 1.0~1.5 的公共端子。图 2-4 上侧的 0.0~0.4 和 0.5~1.1 为输出端子，1M、1L+为输出组 Q0.0~0.4 的电源端子，2M、2L+为输出组 Q0.5~1.1 的电源端子。

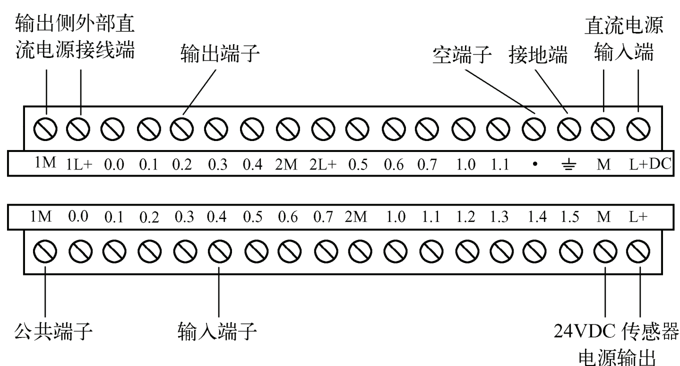


图 2-4　CPU224 DC/DC/DC 的接线端子

2．漏型输入和源型输入

对于 S7-200 PLC 来说，漏型输入是指电流从 PLC 的输入端流进，从公共端流出；源型输入是指电流从 PLC 公共端流进，从输入端流出。S7-200 PLC 的 DC 输入端子在接线时可以按漏型输入连接，也可以按源型输入连接，连接图如图 2-5 和图 2-6 所示。

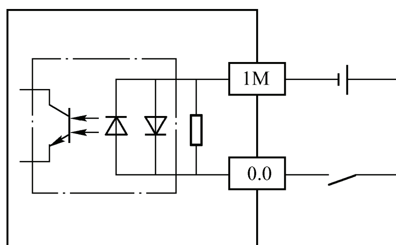


图 2-5　漏型输入

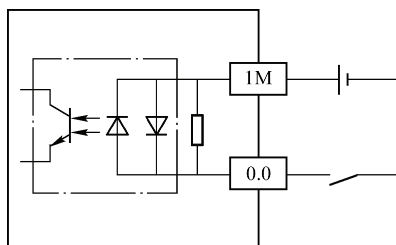


图 2-6　源型输入

3．漏型输出和源型输出

S7-200 PLC 输出有漏型输出和源型输出两种类型，是针对直流负载而言的。漏型输出是指负载电流流入输出端子，而从公共端子流出。源型输出是指负载电流从输出端子流出，而从公共端子流入。对于 S7-200 PLC 来说，一般采用源型输出、集电极开路输出。如图 2-7 所示。

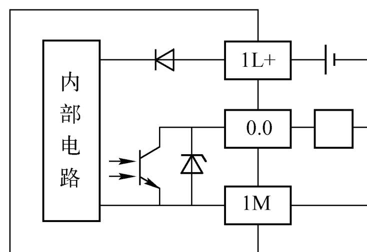


图 2-7　源型集电极开路输出

4．外部接线实例

以 CPU224 DC/DC/DC 型 PLC 为例，如图 2-8 所示，在 PLC 的输入端接入一个按钮 SB1、一个限位开关 SQ1、以及一个接近开关 SP1；输出为一个电磁阀 YV1。

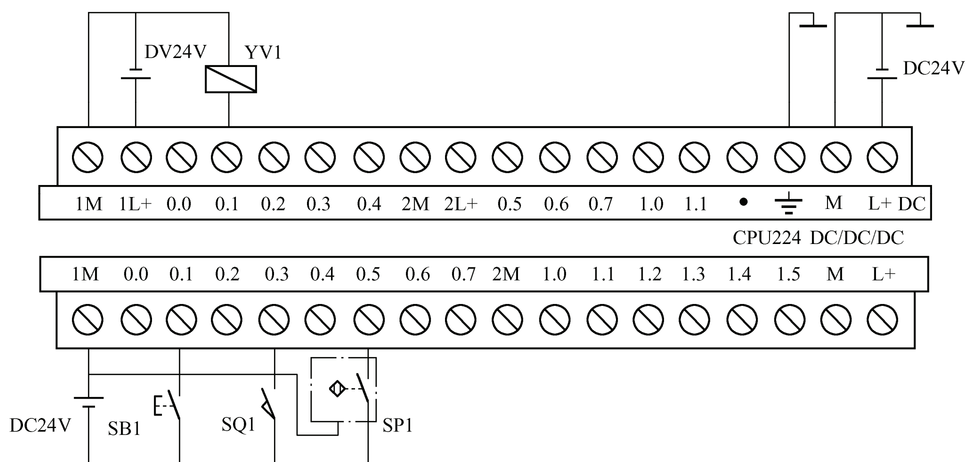


图 2-8　CPU224 DC/DC/DC 外部接线图

2.2　S7-200 PLC 的编程元件

2.2.1　S7-200 PLC 的数据类型及寻址方式

1．数据类型

（1）数据存储类型

S7-200 PLC 的指令参数所用的基本数据类型有 1 位布尔型（BOOL）、8 位字节型（BYTE）、16 位无符号整数（WORD）、16 位有符号整数（INT）、32 位无符号双字整数（DWORD）、32 位有符号双字整数（DINT）、32 位实数型（REAL）。

CPU 存储器中存放的数据类型可分为 BOOL、BYTE、WORD、INT、DWORD、DINT、REAL。不同的数据类型具有不同的数据长度和数值范围。在上述数据类型中，用字节（B）、字（W）、双字（D）分别表示 8 位、16 位、32 位数据的数据长度。不同的数据长度对应的数值范围见表 2-6。例如，数据长度为字（W）型的无符号整数（WORD）的数值范围为 0～65535。不同数据长度的数值所能表示的数值范围是不同的。

表 2-6　数据长度和数值范围

数 据 长 度	无符号整数数值范围		有符号整数数值范围	
	十进制	十六进制	十进制	十六进制
字节 B（8 位）	0～255	0～FF	−128～127	80～7F
字 W（16 位）	0～65,535	0～FFFF	−32,768～32,767	8000～7FFF
双字 D（32 位）	0～4,294,967,295	0～FFFFFFFF	−2,147,483,647~2,147,483,647	80000000～7FFFFFFF
实数 R（32 位）	-10^{38}~10^{38}			
位 BOOL	数值 0 或 1			

SIMATIC 指令集中，指令的操作数具有一定的数据长度。如整数乘法指令的操作数是字型数据；数据传送指令的操作数可以是字节、字或双字型数据。

（2）常数的表示方法

S7-200 系列 PLC 的许多指令都会使用常数，常数的数据长度可以是字节、字或双字。在 CPU 中常数都是以二进制形式来存储的，为表示和使用方便，书写时经常用二进制、十进制、十六进制、ASCII 码或实数等多种形式。书写要求如下：

十进制常数，直接书写，如 3458；十六进制，书写时前面加 16#，如 16#13AC；二进制，书写时前面加 2#，如 2#0001 1011 1110 0001；实数（浮点数），采用科学计数法书写，如 +1.230516E−38、−1.584345E+27。

2．编址方式

S7-200 系列 PLC 为每一种元件分配一个存储区域，并采用不同字母作为区域标识符，如数字量输入，区域标识符为 I；数字量输出，区域标识符为 Q；模拟量输入为 AI；模拟量输出为 AQ；内部标志位存储器为 M；特殊标志位存储器为 SM；变量存储器为 V；局部存储器为 L；顺序控制存储器为 S；T 为定时器；C 为计数器；HC 表示高速计数器。

存储器的单位可以是位、字节、字或双字，所以编址方式相应也分为位、字节、字或双字编址。

3．寻址方式

S7-200 CPU 将程序中的各类信息和数据存储在不同的存储器单元，每个单元都确定一个唯一的地址，CPU 通过地址来访问其对应的数据，称为寻址。

（1）直接寻址

1）位寻址格式：对于 I、Q、M、SM、S、V、L 这些存储器，按位寻址的格式为 Ax.y。其中 A 为存储器区域名称，x 为字节地址，y 为字节内的位地址，例如 I1.5、Q0.1、M10.0、V100.6。对于 T 和 C 存储器，按位寻址格式为 Ax，例如 T0、T100、C0、C56。

2）字节、字和双字寻址格式：对于 I、Q、M、SM、S、V、L 这些存储器，可以按字节、字、双字寻址，寻址格式为 ATx。其中 A 为存储器区域名称，T 的取值分别可以是 B（字节）、W（字）、D（双字），x 为字节地址。例如图 2-9，VB50 表示以字节的方式存取，VW50 表示存取 VB50、VB51 组成的字（VB50 为高 8 位字节、VB51 为低 8 位字节）；VD50 表示存取 VB50~VB53 组成的双字（VB50 为最高 8 位字节，VB53 为最低 8 位字节）。

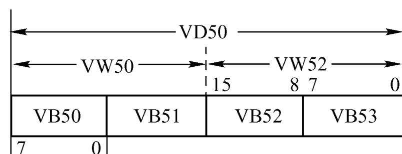


图 2-9　字节/字/双字寻址格式示例

（2）间接寻址

S7-200 系列 PLC 的间接寻址方式是指数据存放在存储器或寄存器中，在指令中只出现所需数据所在单元的内存地址的地址。存储单元的地址的地址又称为地址指针。间接寻址一般在处理内存连续地址中的数据时使用。

S7-200 系列 PLC 可以对 I、Q、V、M、S、T（仅当前值）、C（仅当前值）间接寻址，但不能对独立的位（Bit）或模拟量进行间接寻址。用间接寻址方式存取数据需要做的工作有

3 个，分别是建立指针、用指针存取数据和修改指针。

1）建立指针。使用间接寻址时，首先要创建一个指向该位置的指针。指针为双字（32 位），用于存放另一个存储器的地址，只能用 V、L 或累加器 AC1、AC2 和 AC3 作指针。生成指针时，必须用双字传送指令（MOVD）将数据所在单元的内存地址送入指针，指令的输入操作数开始处加&符号，表示某一存储器位置的地址，而不是存储器里的值，指令的输出操作数为指针地址。

例如：MOVD &VB100，AC1。该指令的含义是将 VB100 的地址（32 位）送入累加器 AC1 中。

2）用指针存取数据。用指针来存取数据时，操作数前面加“*”符号，表示该操作数为一指针。

例如：MOVW *AC1，AC0。该指令的含义是将 AC1 中的内容为起始地址的一个字长的数据（即 VB100，VB101）送入 AC0 的低 16 位，如图 2-10 所示。

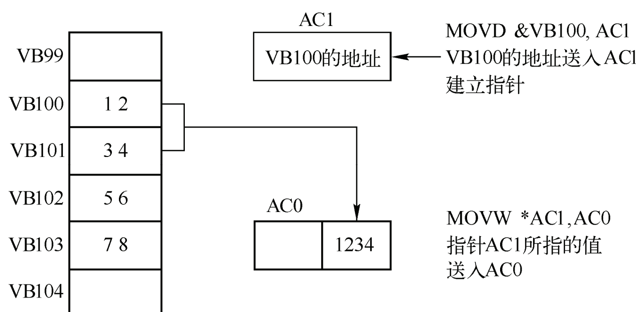


图 2-10　间接寻址方式

3）修改指针。连续存储数据时，通过修改指针很容易存取某一串数据。指针是 32 位的数据，应使用双字指令来修改指针值，如 INCD AC1，将指针加 1。另外修改指针时，要记住访问数据的长度：存取字节时，指针加 1；存取字时，指针加 2；存取双字时，指针加 4。

2.2.2　S7-200 PLC 的编程资源

1．输入继电器（I）

输入继电器一般都有一个 PLC 的输入端子与之对应，它是 PLC 用来接收工业现场各设备输入信号的接口，其状态仅取决于输入端开关元件的状态。当接在输入端子的开关元件闭合时，输入继电器的线圈得电，在程序中的动合触点闭合，动断触点断开；这些触点可以在编程时任意使用，使用次数不受限制。

PLC 输入端子可连接外部的动合（NO）触点或动断（NC）触点，输入端连接不同触点，其内部软元件对应的状态也相应不同。编程时应注意的是，输入继电器的线圈只能由外部信号来驱动，不能在程序内用指令来驱动，因此在编制的梯形图中只能出现输入继电器的触点，而不应出现输入继电器的线圈，其触点也不能直接输出驱动负载。

输入继电器有 IB0~IB15 共计 16 个字节，按位存取的地址范围是 I0.0~I15.7。

2．输出继电器（Q）

输出继电器也有一个 PLC 的输出端子与之对应，它是用来将输出信号传送到负载的接口，用于驱动负载。当输出继电器的线圈得电时，对应的输出端子接通，负载电路开始工作。每一个输出继电器的动合触点和动断触点在编程时，可不限次数使用。

编程时需要注意的是，外部信号无法直接驱动输出继电器，它只能在程序内部驱动。

输出继电器有 QB0 ~ QB15 共计 16 个字节，按位存取的地址范围是 Q0.0 ~ Q15.7。

3．辅助继电器

（1）通用辅助继电器（M）

PLC 内部有很多辅助继电器，和输出继电器一样，只能由程序驱动，每个辅助继电器也有无数对动合、动断触点供编程使用。辅助继电器的触点在 PLC 内部编程时可以任意使用，但它不能直接驱动负载，外部负载必须由输出继电器的触点来驱动。

在逻辑运算中经常需要一些辅助继电器作为辅助运算用，这些器件往往用作状态暂存、移位等运算。另外一些辅助继电器还有一些特殊功能。

辅助继电器可以按位、字节、字或双字来存取，按位存取的地址范围是 M0.0~M31.7，共计 32 个字节。

（2）特殊辅助继电器（SM）

特殊辅助继电器，又称特殊标志位存储器。对于不同的 CPU，特殊标志位存储器的地址范围有所不同，CPU224 的地址范围为 SM0.0~SM179.7，可按位、字节、字和双字来存取。每一个特殊标志位存储器都有特定的功能，常用特殊辅助继电器元件见表 2-7。

表 2-7　S7-200 系列 PLC 的常用特殊辅助继电器元件

编　　号	功 能 描 述
SM0.0	RUN 状态监控；PLC 为 RUN 运行状态时，该位始终为 ON
SM0.1	首次扫描时为 1；PLC 由 STOP 转为 RUN 状态时，该位接通一个扫描周期，用于程序的初始化
SM0.2	如果保留性数据丢失，该位为一次扫描循环打开，可用作错误内存位或激活特殊启动顺序的机制
SM0.3	开机后进入 RUN 模式，该位被置 ON 一个扫描周期，可用于在启动操作之前提供机器预热时间
SM0.4	1min 时钟脉冲；该位输出一个占空比为 50％的分时钟脉冲
SM0.5	1s 时钟脉冲；该位输出一个占空比为 50％的秒时钟脉冲
SM0.6	该位是扫描循环时钟，为一次扫描打开，然后为下一次扫描关闭，可用于扫描计数器输入
SM0.7	该位表示 PLC“模式”开关的位置（1=运行；0=终止），当开关位于运行位置时，可通过该位启用自由端口模式
SM1.0	零标志位；运算结果为 0 时，该位置 1
SM1.1	溢出标志位；运算结果溢出或查出非法数值时，该位置 1
SM1.2	负数标志位；数学运算结果为负时，该位为 1

4．顺序控制继电器（S）

顺序控制继电器常用于顺序控制或步进控制中，用其表示某一特定的操作或程序步，并与其指令（SCR）一起使用实现顺序或步进控制功能流程图（SFC）的编程。顺序控制继电器的线圈只能由程序指令驱动，触点在 PLC 内部编程时可以任意使用。

顺序控制继电器存取的地址范围是 S0.0~S31.7。

5．定时器（T）

PLC 提供的定时器相当于继电器控制系统中的时间继电器，是累计时间增量的编程元件，定时值由程序设置。每个定时器都对应一个 16 位的当前值寄存器，当定时器的输入条件满足时开始计时，当前值从 0 开始按一定的时间单位增加，当定时器的当前值等于程序中的设定值时，定时时间到，定时器的触点动作。每个定时器提供的动合触点和动断触点有无数个。

CPU222、CPU224 及 CPU226 的定时器地址范围是 T0~T255，用定时器地址（T 和定时器号，如 T0）来存取当前值和定时器位。

6．计数器（C）

计数器用于累计计数输入端接收到的由断开到接通的脉冲个数，其设定计数值由程序设置。计数器的当前值是 16 位或 32 位有符号整数，用于存储累计的脉冲个数，当计数器的当前值等于设定值时，计数器的触点动作。每个计数器提供的动合触点和动断触点有无数个。

计数器的地址范围是 C0~C255，用计数器地址（C 和计数器号，如 C0）来存取当前值和计数器位。

7．高速计数器（HC）

高速计数器的地址范围根据 CPU 的型号有所不同，CPU221、CPU222 各有 4 个高速计数器，分别是 HC0~HC3，CPU224、CPU226 各有 6 个高速计数器，分别是 HC0~HC5。

8．数据存储器

（1）变量存储器（V）

变量存储器主要用于存储变量。它可以存放程序执行过程中的中间变量，也可以保存与工序或任务相关的变量。在进行数据处理时，变量存储器会被经常使用。

变量存储器存取的地址范围根据 CPU 的型号有所不同，CPU221、CPU222 为 V0.0~V2047.7，CPU224、CPU226 为 V0.0~V5119.7。

（2）局部变量存储器（L）

局部变量存储器和变量存储器相类似，也是用来存储变量。区别在于局部变量存储器主要用来存放局部变量，而变量存储器存放的是全局变量。S7-200 系列 PLC 有 64 个字节的局部存储器，PLC 运行时，根据需要动态地分配局部变量存储器，其中 60 个字节可以作为暂时存储器，或给子程序传递参数，另外 4 个字节作为系统的保留字节。

局部变量存储器存取的地址范围是 L0.0~L63.7。

（3）累加器（AC）

累加器是用来暂存数据的寄存器，它可以用来存放运算数据、中间数据和结果，也可以用来向子程序传递参数或从子程序返回参数。

S7-200 系列 PLC 提供了 4 个 32 位的累加器，地址范围是 AC0~AC3。

2.3 编程软件和程序结构

1．编程软件和编程语言

STEP 7-Micro/WIN 是西门子公司专为 SIMATIC S7-200 系列可编程序控制器研制开发的编程软件，是基于 Windows 操作平台的应用软件，功能强大，既可用于开发用户程序，又可实时监控用户程序的执行状态。

在进行编程设计前，应采用 PC/PPI 电缆将计算机的 RS-232 接口与 S7-200 PLC 的 RS-485 端口连接起来。

所谓程序编制，就是用户根据控制对象的要求，利用 PLC 厂家提供的程序编制语言，将一个控制要求描述出来的过程。在 STEP 7-Micro/WIN 编程软件中，提供了梯形图、语句表和功能块三种编程语言供用户选用，且三种语言可相互转换。

2．程序结构

在进行 PLC 程序设计时，应根据所要完成的控制任务的难易程度来确定程序结构。对于较为简单的控制任务，PLC 程序只要单一程序块即可满足；而对于较为复杂的控制任务，可将程序分解为模块化结构，便于程序的编写和录入；一般模块化程序结构由主程序、子程序和中断处理程序组成。

（1）主程序

主程序是 PLC 程序的主体，每个程序中有且只有一个主程序，主程序控制整个程序的执行，S7-200 在每个扫描周期中都会顺序执行主程序中的指令，主程序主要完成参数初始化、调用子程序和中断程序等。

（2）子程序

子程序是程序的可选单元，只有在调用时才会执行；一般可通过主程序、中断程序或另一子程序调用执行。如当需要重复执行某段程序时，采用子程序结构是非常有用的，将需重复执行的程序写在子程序里，当需使用时通过子程序调用。合理的子程序可以优化程序结构、减少程序长度和缩短扫描时间。

（3）中断程序

中断程序也是程序的可选单元，当特定的中断事件发生时，程序会转而执行中断程序；S7-200 中断程序类型有通信端口中断、高速计数器中断、脉冲输入输出的上升沿/下降沿中断以及时基中断等。中断程序用来处理不能事先预测的事件，或者与用户程序执行时序无关的操作。

2.4 STEP 7-Micro/WIN 编程软件的使用

启动 STEP 7-Micro/WIN 编程软件，用鼠标双击桌面上 STEP 7-Micro/WIN 编程软件的快捷图标，进入编程主界面。

2.4.1 主界面的认识

图 2-11 所示为 STEP 7-Micro/WIN 主界面（以 V4.0 SP6 版本为例），界面包括菜单栏、工具栏、浏览栏、指令树窗口、程序编辑器窗口和状态栏等部分。

1）“浏览栏”提供快速切除用户窗口的控制按钮，包括“查看”和“工具”两部分，每一部分又包括若干按钮，单击任一按钮，可打开此按钮对应的窗口。

2）“指令树”提供一个树型查看方式，包括项目和指令两部分内容。启动软件自动生成项目 1，默认程序块中包括主程序、子程序和中断程序；通过选择指令部分的内容，可以在程序编辑窗口编辑用户程序。

3）“菜单栏”包括八个主菜单项：文件、编辑、查看、PLC、调试、工具、窗口和帮助。

菜单栏提供了使用鼠标或键盘进行操作的主要方式。

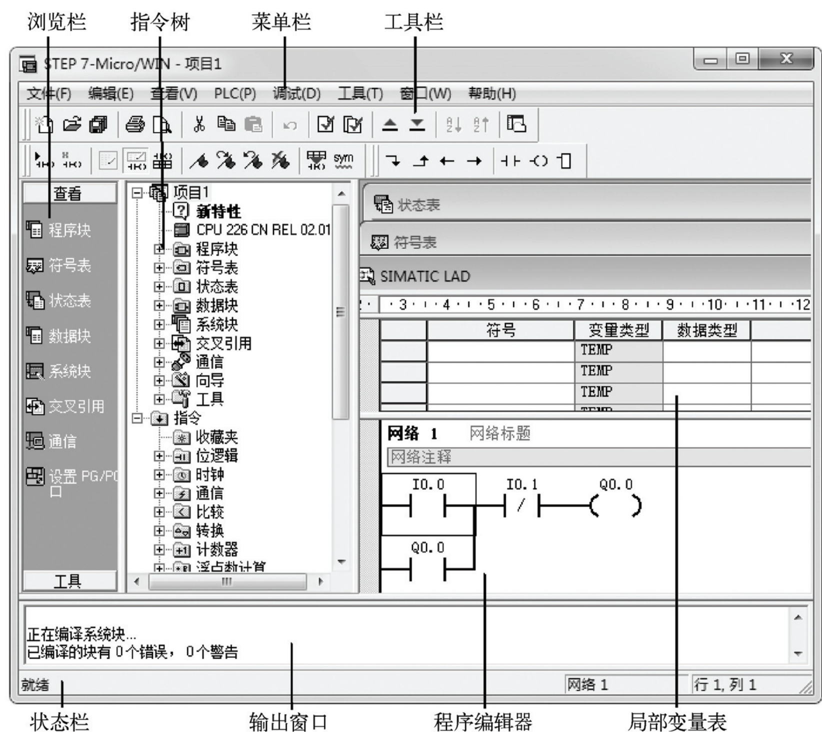


图 2-11　STEP 7-Micro/WIN 主界面

4）“工具栏”将最常用的操作以按钮形式设定到主窗口，提供给用户最方便的鼠标访问，用户可以根据需要和习惯定制工具栏的内容和外观，方法如下：单击“查看”→“工具栏”→选中的工具组将出现在“工具栏”中。

5）“状态栏”用来提供执行 STEP 7-Micro/WIN 的状态信息，显示当前操作的信息。

6）“输出窗口”用来显示程序编译的结果信息。

7）“程序编辑器”可以用梯形图、语句表或功能图程序编辑器编写和修改用户程序，编程工作主要在程序编辑窗口中完成。

8）“局部变量表”如图 2-12 所示，每个程序块都对应一个局部变量表，在带参数的子程序调用中，参数的传递就是通过局部变量表进行的。



图 2-12　局部变量表

2.4.2　程序编辑

1．生成程序文件

程序文件的来源有三个：打开已有的程序文件、从 PLC 上载程序文件以及新建一个程序文件。

（1）打开程序文件

打开磁盘中已有的程序文件。可用“文件”菜单中的“打开”命令，或单击工具栏中的“打开”按钮，选择已编辑好的程序文件。

（2）上载程序文件

在与 PLC 建立通信的情况下，可以将存储在 PLC 中的程序和数据传送给计算机。可用“文件”菜单中的“上载”命令，或单击工具栏中的“上载”按钮，弹出图 2-13 所示对话框，单击“上载”按钮，PLC 的程序即下载到计算机中。

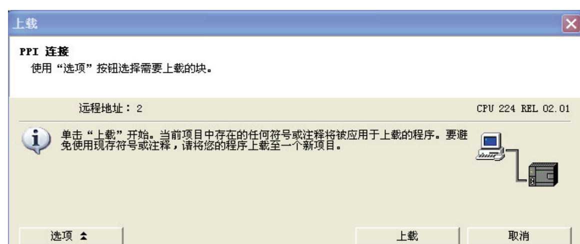


图 2-13　PLC 上载程序

（3）新建程序文件

可以选择“文件”菜单中的“新建”选项或单击工具栏中的“ ”按钮创建一个新程序文件。现以新建程序文件为例，简单介绍程序的生成。

1）确定 PLC 的 CPU 型号：鼠标右键单击“Project1”→“CPU221”图标，在弹出的下拉菜单中单击“类型”，弹出图 2-14 所示“PLC 类型”对话框，选择实际的 PLC 型号和版本；也可用“PLC”菜单中的“类型”项来选择 PLC 型号。若 PLC 与计算机之间已经成功地建立了通信连接，单击对话框中的“读取 PLC”，可以通过通信读出 PLC 的类型和 CPU 版本号。

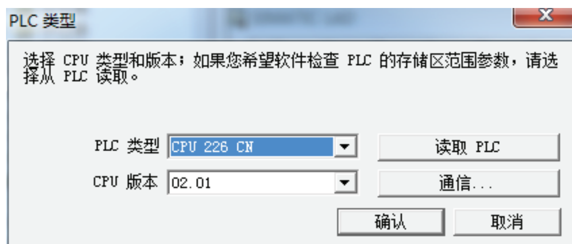


图 2-14　“PLC 类型“对话框

2）选择菜单命令“工具”→“选项”，弹出相应的选项窗口。在窗口的“常规”选项中可以选择语言、默认的编辑器及编程模式等项目。如图 2-15 所示，选择“General”→“Chinese”，单击“OK”，编程页面关闭，重新打开编程软件，软件为中文显示界面。

3）程序更名：如果要更改程序的文件名，可单击“文件”菜单中“另存为”项，在弹出的对话框中键入新的文件名。程序块中主程序的名称一般用默认名称，任何程序文件都只有一个主程序。对子程序和中断程序的更名可在指令树窗口中右键单击需要更名的子程序或中断程序名，在弹出的下拉菜单中单击“重命名”，然后键入新名称。

4）添加子程序或中断程序。

方法 1：在指令树“项目”窗口中右键单击“程序块”图标，在弹出的选择按钮中单击“插入子程序”或“插入中断程序”项。

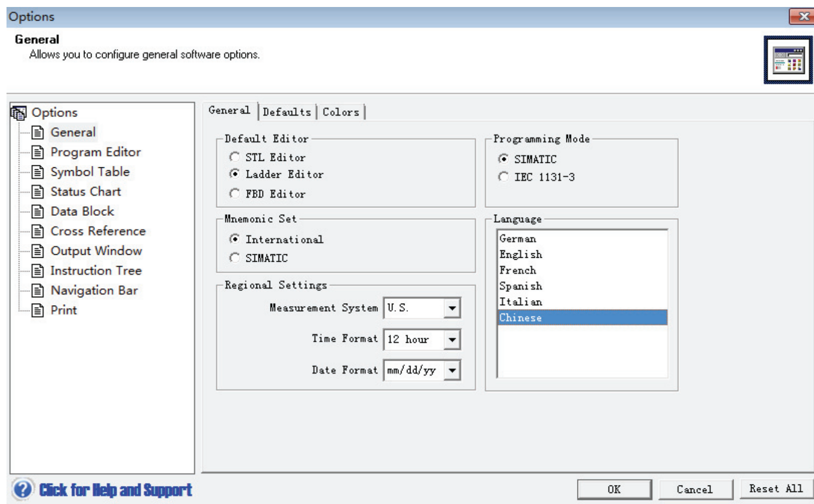


图 2-15　修改编程语言

方法 2：用“编辑”菜单中“插入”项下的“子程序”或“中断程序”来实现。

方法 3：鼠标右键单击编辑窗口，在弹出的选项中选择“插入”项下的“子程序”或“中断程序”命令。新生成的子程序或中断程序会根据已有的子程序或中断程序的数目自动递增编号，用户可将其更名。如图 2-16 所示。



图 2-16　在编辑窗口添加子程序或中断程序文件

2．编译程序

程序文件编辑完成后，可用“PLC”菜单中的“编译”命令，或工具栏中的“编译”按钮进行离线编译。编译结束后，将在输出窗口中显示编译结果。

3．下载程序

程序只有在编译正确后才能下载到 PLC 中。下载前，PLC 必须处于“STOP”状态。如果不在 STOP 状态，可单击工具栏中“停止”按钮，或选择“PLC”菜单中的“停止”命令，如图 2-17 所示；也可以将 CPU 模块上的方式选择开关直接扳到“STOP”位置。为了使下载的程序能正确执行，下载前也可以将 PLC 中存储的原程序清除；单击“PLC”菜单项中的“清除”命令，在出现的对话框中选择“清除全部”即可。

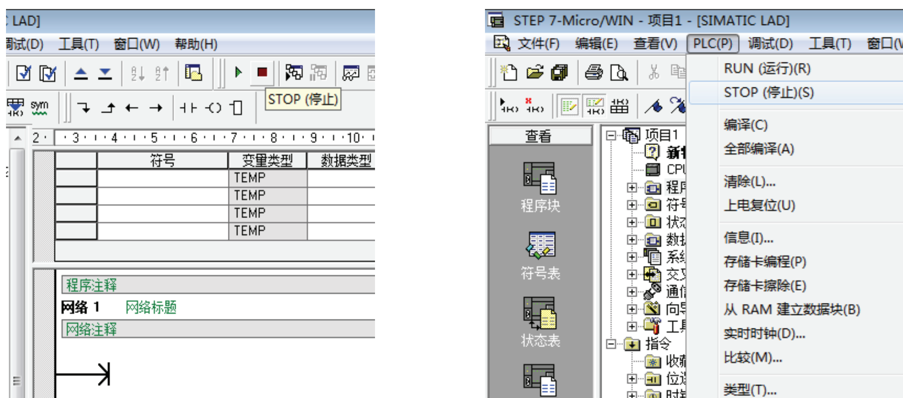


图 2-17　PLC 停止状态的远程操作方法

4．运行模式下编辑程序

在运行模式下，可以对用户程序作少量修改，而不必转换到停止模式。修改后的程序一旦下载将立即影响系统的运行，操作如下：

1）在运行模式下，选择“调试”菜单中“RUN（运行）模式下程序编辑”命令。如果编程软件当前打开的项目与 PLC 中的程序不同，将提示上载 PLC 中的程序，该功能只能编辑 PLC 中的程序。进入 RUN 模式编辑后，光标会变成带有 PLC 的图标。

2）编辑程序前，应该退出程序监控状态。

3）程序编译成功后，可用“文件”菜单中的“下载”命令，或单击工具栏中的“下载”按钮将程序下载到 PLC 中。

4）若要退出运行模式编辑，需要再选择一次“调试”菜单中“RUN（运行）模式下程序编辑”命令。

2.4.3　程序的调试及监控

STEP 7-Micro/WIN 编程软件允许用户在软件环境下直接调试并监控程序的运行。

1．用状态表监控程序

STEP 7-Micro/WIN 编程软件可以使用状态表来监视用户程序的执行情况，并可对编程元件进行强制操作。

使用状态表：在“浏览栏”中单击“状态表”图标，或使用“调试”菜单中的“状态表”命令就可打开状态表窗口，如图 2-18 所示。在状态表的“地址”栏中键入要监控的编程元件的直接地址（或用符号表中的符号名称），在“格式”栏中显示编程元件的数据类型，在“当前值”栏中可读出编程元件的状态和当前值。

状态表

	地址	格式	当前值	新值
1	I0.1	位	2#1	
2	T37	位	2#1	
3	Q0.0	位	2#1	
4		有符号		
5		有符号		

图 2-18　“状态表”窗口

工具栏中状态表的编辑工具有顺序排序、逆序排序、单次读取、全部写、强制、解除强制、解除所有强制以及读所有强制等。

强制操作是指对状态表中的变量进行强制性赋值，只有在 PLC 处于 RUN 模式时，才会显示强制状态，如图 2–18 中的 I0.1 为强制状态。

2．梯形图程序的状态监控

选择执行菜单命令“调试”→“开始程序状态监控”后，进入程序的监控状态。在这种模式下，只有在 PLC 处于 RUN 模式时才刷新程序段中的状态值；在 RUN 模式下启动程序状态监控后，将用不同的颜色显示出梯形图中各元件的状态。

图 2–19 为程序的在线监控状态。其中 I0.1 元件工作在 ON 状态，其触点中间为实心矩形（实心矩形为蓝色）； I0.0 工作在 OFF 状态，则触点为空心；定时器处在工作状态下，则方框颜色为绿色，表明其中包含有效数据；由于 Q0.0=1，所以方框指令 MOV_B 被执行，方框颜色变为蓝色；若指令执行时发生错误则方框颜色变为红色。

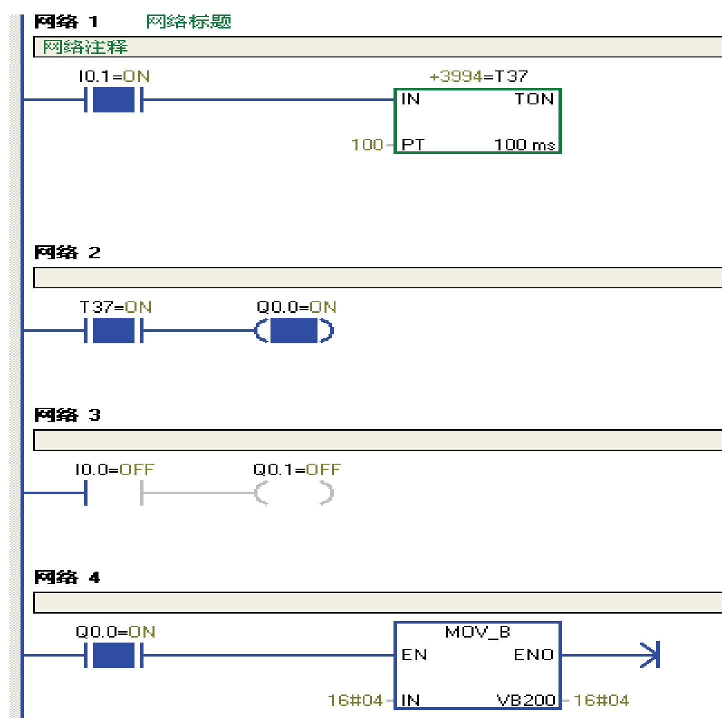


图 2–19　程序在线监控

第 3 章　S7-200 PLC 的指令系统

3.1　位逻辑指令

S7-200 PLC 的位逻辑指令包括触点指令、输出指令、RS 触发指令以及逻辑堆栈指令等，这些指令是构成逻辑控制的最基础指令。

3.1.1　标准触点指令

标准触点指令如图 3-1 所示，在语句表中分别用 LD、A、O 指令表示开始、串联和并联的动合触点；分别用 LDN、AN、ON 表示开始、串联和并联的动断触点；用“＝”表示线圈的输出指令。在梯形图中，触点符号中的“/”表示常闭，标准触点指令中变量的数据类型为 BOOL 型。

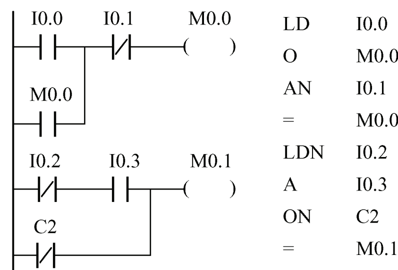


图 3-1　标准触点指令

在梯形图中，当对应存储器地址位为 1 时，动合触点接通，动断触点断开；当对应存储器地址位为 0 时，动合触点断开，动断触点接通。

【例 3-1】 试画出图 3-2a 中 Q0.0 的动作时序图。

分析：在第一个扫描周期，由于 Q0.0 的初始状态为 OFF，Q0.0 的动断触点接通，因此 Q0.0 线圈得电，输出状态为“1”；在第二个扫描周期，由于 Q0.0 的状态为 ON，Q0.0 的动断触点断开，因此 Q0.0 线圈失电，输出状态为“0”；以后将重复上述转换过程，其动作时序图如图 3-2b 所示。



图 3-2　例 3-1 图

3.1.2　取反指令

取反指令 NOT 将它左边电路的逻辑运算结果取反。运算结果若为 1 则变为 0；若为 0 则

变为 1。如图 3-3 所示，当 I0.5=1 时，Q0.0=0，当 I0.5=0 时，Q0.0=1；当 I0.6=1 且 I0.7=0 时，Q0.1=0，否则 Q0.1=1。

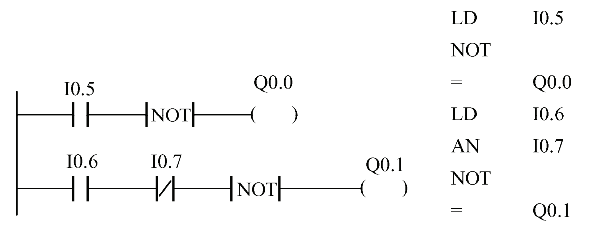


图 3-3 取反指令

3.1.3 边沿脉冲指令

边沿脉冲指令的功能是检测输入信号的上升沿和下降沿，若检测到边沿触发信号，则产生一个扫描周期的脉冲信号。指令格式为：EU，上升沿；ED，下降沿。如图 3-4 所示。

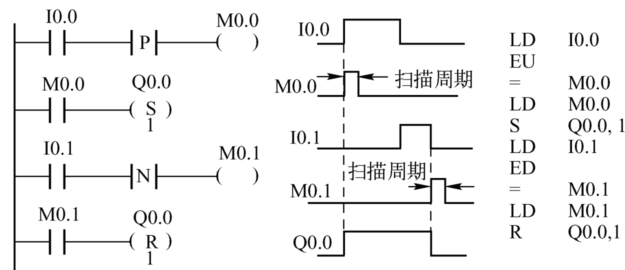


图 3-4 边沿脉冲指令

3.1.4 置位/复位指令

置位指令（S）的功能是将位存储区的某一位开始的 N 个（N=1~255）同类存储器位置 1；复位指令（R）的功能是将位存储区的某一位开始的 N 个（N=1~255）同类存储器位置 0。如图 3-5 所示。

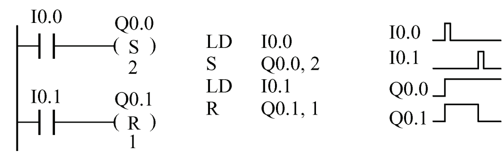


图 3-5 置位、复位指令

【例 3-2】 分析梯形图 3-6a 实现的功能，并画出对应的时序图。

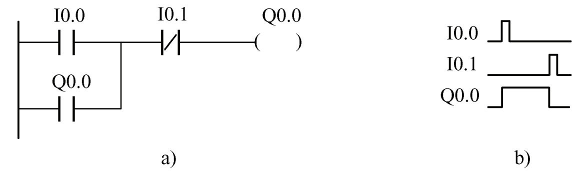


图 3-6 例 3-2 图

分析：图 3-6a 中的启动信号 I0.0 和停止信号 I0.1 持续为 ON 的时间一般都很短，这种信号称为短信号，如何使线圈 Q0.0 保持接通状态呢？利用自身的动合触点使线圈保持通电即“ON”状态的功能称为自锁或自保持功能，自保持控制电路常用于按钮等无机械锁定开关的启停控制。

当启动信号 I0.0 变为 ON 时，I0.0 的动合触点接通，如果这时 I0.1 为 OFF 状态，I0.1 的动断触点接通，Q0.0 的线圈通电，其动合触点接通；放开启动按钮，I0.0 变为 OFF，其动合触点断开，“能流”从左母线经 Q0.0 的动合触点、I0.1 的动断触点流过 Q0.0 的线圈，Q0.0 仍为 ON。

当 I0.1 为 ON 时，它的动断触点断开，停止条件满足，Q0.0 线圈失电，其动合触点断开，即使放开停止按钮使 I0.1 的动断触点恢复接通状态，Q0.0 的线圈仍然断电，对应的时序图如图 3-6b 所示。这种自保持的功能与图 3-6 用 SET/RST 指令实现的功能一样，它们的输入输出信号有相似的时序图。

3.1.5 立即指令

立即指令的功能是对 I/O 点进行快速的存取，不受 PLC 扫描周期的影响，可以提高 PLC 对外部设备的反应速度。LDI 立即取指令，LDNI 立即取反指令，OI 立即或指令，ONI 立即或反指令，AI 立即与指令，ANI 立即与反指令，＝I 立即输出指令，SI 立即置位指令，RI 立即复位指令。如图 3-7 所示。

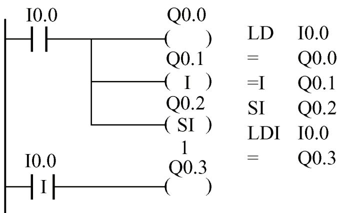


图 3-7 立即指令

3.1.6 逻辑电路块的连接指令

电路块连接指令有 2 个，分别是 OLD、ALD。如图 3-8 所示。

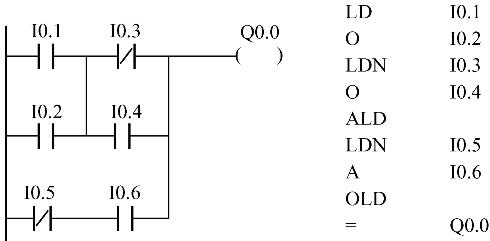


图 3-8 电路块连接指令

OLD 是指将两个以上触点串联形成的电路块并联起来；ALD 是指将两条以上支路并联形成的电路块串联起来；OLD、ALD 指令是独立指令，使用时不带软元件号。

3.1.7 堆栈操作指令

S7-200 有一个 9 位的堆栈，堆栈按“先进后出，后进先出”的原则存取，即先入的数据放在栈下方，后进的数据放在栈上方，出栈时上方的数据（后进的数据）先出去，下方的数据（先进的数据）后出去。栈顶用来存储逻辑运算的结果，下面的 8 位用来存储中间结果。与堆栈有关的指令有 4 个。

入栈指令（LPS）：复制栈顶的值并将其压入堆栈的下一层，栈中原来的数据依次向下一层推移，原栈底值被推出丢失。

读栈指令（LRD）：将堆栈中第 2 层的数据复制到栈顶，2~9 层的数据不变，但原栈顶值丢失。

出栈指令（LPP）：使栈中各层的数据向上移动一层，第 2 层的数据成为堆栈的新的栈顶值，原栈顶值被推出丢失。

装载堆栈（LDS）：LDS n，n=1~8。复制堆栈内第 n 层的值到栈顶，栈中原来的数据依次向下移一层，原栈顶值丢失。

这些堆栈操作指令如图 3-9 所示，其中 LPS 和 LPP 必须成对使用，主要用在多重分支输出回路。

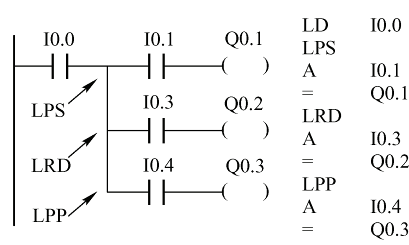


图 3-9　堆栈操作指令

【例 3-3】 分析梯形图 3-10a，并写出对应指令表。

分析：对于只有一个分支的梯形图，只能使用 LPP，不能使用 LRD 指令。当 I0.0=1 且 I0.1=1 时，则 Q0.0=1；当 I0.0=1 且 I0.2=1 时，则 Q0.1=1。对应指令表如图 3-10b 所示。

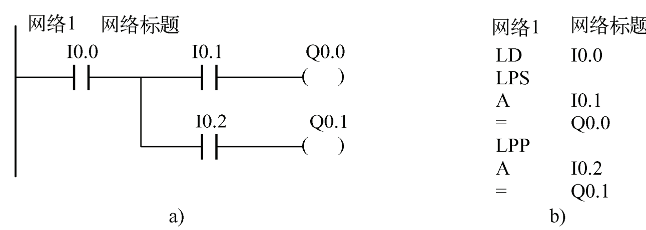


图 3-10　例 3-3 图

【例 3-4】 分析梯形图 3-11a 的结构，并写出对应指令表。

分析：图 3-11a 梯形图的分支电路上又有分支存在，是个两层堆栈，栈指令 LPS、LPP 要求成对出现，即用几次进栈指令就需要应用几次出栈指令，指令表如图 3-11b 所示。

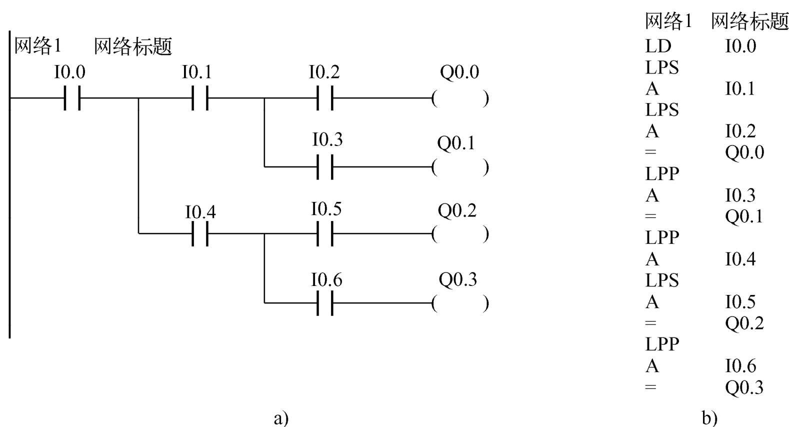


图 3-11　例 3-4 图

3.2　定时器指令

S7-200 系列 PLC 的定时器共有 3 种类型：接通延时定时器 TON、断开延时定时器 TOF 和有记忆接通延时定时器 TONR；定时器有 1ms、10ms、100ms 三种分辨率，分辨率取决于定时器号，见表 3-1；其中 TON 与 TOF 共享同一组定时器号，不能重复使用。

表 3-1　S7-200 系列 PLC 定时器

类　　型	分辨率/ms	定时范围/s	定 时 器 号
TONR	1	32.767	T0、T64
	10	327.67	T1~T4、T65~T68
	100	3276.7	T5~T31、T69~T95
TON、TOF	1	32.767	T32、T96
	10	327.67	T33~T36、T97~T100
	100	3276.7	T37~T63、T101~T255

1．接通延时定时器指令

接通延时定时器 TON 的工作过程如图 3-12 所示，当定时器 T37 的 IN 输入端 I0.0 变为 ON 时，定时器 T37 开始定时；如果当前值在小于预置时间（即 PT 端的设定值 50）时，I0.0 变为 OFF，则定时器 T37 的当前值恢复为 0；如果当前值大于等于设定值 50 时（即定时时间达到 5s），定时器的动合触点 T37 为 ON，Q0.0 线圈得电。到达设定值后，当前值仍继续计数，直到最大值 32767。当输入电路断开时，定时器自动复位，当前值被清零，动合触点 T37 为 OFF。

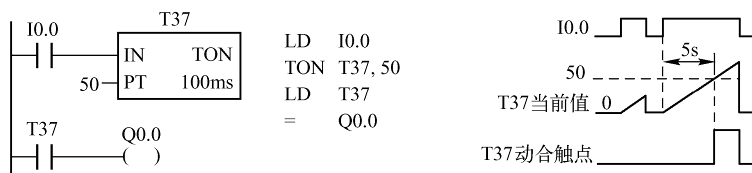


图 3-12　接通延时定时器指令

2．断开延时定时器指令

断开延时定时器 TOF 的工作过程如图 3-13 所示，当定时器 IN 输入端 I0.0 为 ON 时，定时器的动合触点 T39 闭合，当前值被清零；当 I0.0 由 ON 变为 OFF 时，定时器 T39 开始定时，定时时间达到设定值 5s 时，定时器动合触点 T39 变为 OFF，定时器的当前值保持不变，直到输入电路 I0.0 的状态再次为 ON。

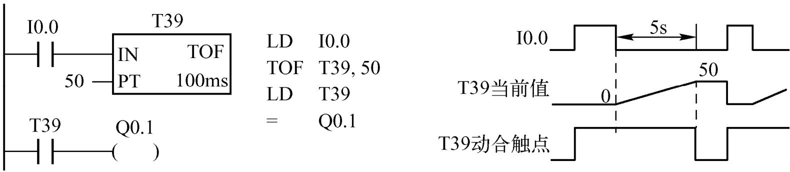


图 3-13 断开延时定时器指令

TOF 与 TON 不能共享相同的定时器号，例如不能在同一段程序中同时对 T37 使用指令 TON 和 TOF，否则会出现逻辑混乱的现象。

3．保持型接通延时定时器指令

保持型接通延时定时器 TONR 的工作过程如图 3-14 所示，其工作原理与接通延时定时器 TON 相类似，在输入电路接通时，开始定时。当前值大于等于 PT 端指定的设定值时，定时器的动合触点变为 ON，达到设定值后，仍然继续计数，直到最大值 32767；不同的是保持型接通延时定时器在输入电路断开时，当前值保持不变。

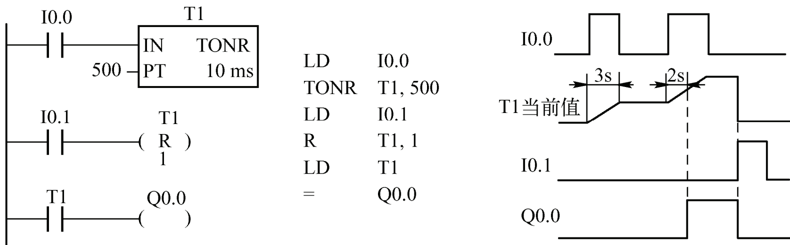


图 3-14 保持型接通延时定时器指令

3.3 计数器指令

计数器用来累计输入脉冲的个数。S7-200 系列 PLC 的计数器有 3 种类型：加计数器 CTU、减计数器 CTD 和加减计数器 CTUD。计数器的编号范围是 C0~C255，不同类型的计数器不能使用同一计数器号。

1．加计数器

加计数器的工作过程如图 3-15 所示，当计数器 C0 的复位输入端（R）电路断开，加计数脉冲输入（CU）I0.3 有上升沿，则计数器的当前值加 1，直至计数最大值 32767。当前值大于等于设定值（PV）4 时，计数器的动合触点 C0 变为 ON。当复位输入（R）I0.4 为 ON 或执行复位指令时，计数器被复位，计数器变为 OFF，当前值被清零。

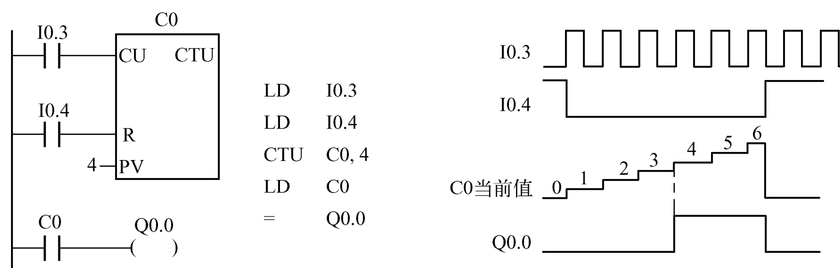


图 3-15　加计数器指令

2．减计数器

减计数器的工作过程如图 3-16 所示，在减计数器 C1 脉冲输入（CD）I0.3 的上升沿，从设定值开始，计数器 C1 的当前值减 1，减至 0 时，计数器 C1 的动合触点变为 ON。装载输入（LD）I0.4 为 ON 时，计数器 C1 被复位，并把设定值装入当前值。

3．加减计数器

加减计数器的工作过程如图 3-17 所示，在加计数器脉冲输入（CU）I2.1 的上升沿，计数器 C2 的当前值加 1，在减计数器脉冲输入（CD）I2.2 上升沿，计数器 C2 的当前值减 1，计数器 C2 的当前值大于等于设定值（PV）3 时，计数器 C2 的动合触点变为 ON。当复位输入（R）I2.3 为 ON 时，计数器 C2 被复位，当前值被清零。

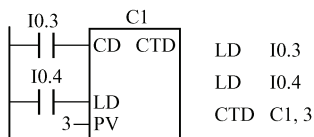


图 3-16　减计数器指令

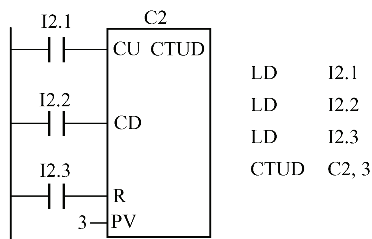


图 3-17　加减计数器指令

3.4　程序控制指令

1．跳转指令

跳转指令的功能是使程序可以根据条件转到同一程序中的具体标号处，该指令包括两个指令：跳转指令（JMP）和标号指令（LBL）。跳转指令（JMP）必须和标号指令（LBL）配合使用，当跳转指令（JMP）的条件为真时，程序从当前位置直接跳转到标号指令（LBL）所指向的位置继续执行，标号指令是跳转指令的目标标号。

如图 3-18 所示，当 I0.0 为 ON 时，程序跳离对 Q0.2 的控制，转而去执行标号下面的程序，Q0.2 保持跳转前的状态，即跳转后 Q0.2 与 I0.1 的状态无关。

2．子程序指令

S7-200 的控制程序由主程序、子程序和中断程序组成。使用子程序可以更好地组织程序结构，如将程序分成若干个功能的小块，便于程序的查错、维护和阅读。

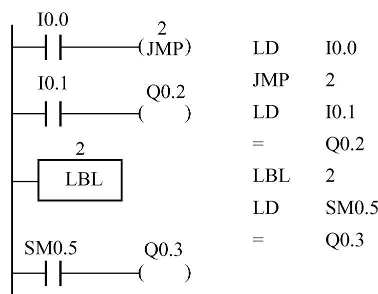


图 3-18　跳转指令

子程序指令包括 3 个指令：建立子程序、子程序调用和子程序返回。建立子程序是通过软件实现的。子程序调用指令的功能是将主程序的执行权转移到子程序 SBR_0 中去。子程序返回分为无条件返回和有条件返回，无条件返回指令是编程软件给子程序自动添加的，当子程序执行结束时，自动返回；有条件返回是指在子程序中用触点控制 RET 指令，触点电路接通时，子程序被终止。

如图 3-19 所示，当 I0.0 闭合时，主程序调用子程序 SBR_0，当 I0.1 闭合时子程序有条件返回；若 I0.0 断开，则子程序中各个元件状态将保持退出子程序之前的状态。

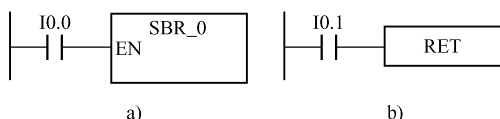


图 3-19　子程序指令

a) 子程序的调用　b) 子程序的返回

3．中断指令

中断程序不是由主程序调用，而是在中断事件发生时由操作系统调用。在执行主程序时，如果有中断信号进来，则停止执行主程序，转而去执行中断程序，执行完中断程序之后，再返回执行主程序。中断技术主要应用于信号采集、实时处理和通信等方面。

在 S7-200 系列中，最多可以有 34 个中断源，每个中断源都分配了一个中断号加以识别。这些中断源大致可以分成三种类型：通信中断、输入/输出中断和时基中断。通信中断是指 PLC 的通信工作在自由通信口模式，在这种模式下可以通过程序来控制通信口；输入/输出中断包括外部输入中断、高速计数器中断和脉冲输出中断；时基中断包括定时中断和定时器中断。

在中断系统中，不同中断源产生的中断有不同的顺序，即优先级。在 S7-200 系列中，优先级最高的是通信中断，其次是输入/输出中断，最后是时基中断。一旦中断被响应之后，中断程序将一直被执行到结束为止，就算有更高级别的中断产生也不会终止正在执行的中断程序。具体的中断优先级及中断事件号见表 3-2。

表 3-2　中断事件及优先级

组 级 优 先	组 内 类 型	中断事件号	中断事件描述	组内优先级
通信中断	通信口 0	8	通信口 0：接收字符	0
		9	通信口 0：发送完成	0
		23	通信口 0：接收信息完成	0

（续）

组级优先	组内类型	中断事件号	中断事件描述	组内优先级
通信中断	通信口 1	24	通信口 1：接收信息完成	1
		25	通信口 1：接收字符	1
		26	通信口 1：发送完成	1
输入/输出中断	脉冲输出	19	PTO0 脉冲输出完成中断	0
		20	PTO1 脉冲输出完成中断	1
	外部输入	0	I0.0 上升沿中断	2
		2	I0.1 上升沿中断	3
		4	I0.2 上升沿中断	4
		6	I0.3 上升沿中断	5
		1	I0.0 下降沿中断	6
		3	I0.1 下降沿中断	7
		5	I0.2 下降沿中断	8
		7	I0.3 下降沿中断	9
	高速计数器	12	HSC0 当前值等于预设值中断	10
		27	HSC0 输入方向改变中断	11
		28	HSC0 外部复位中断	12
		13	HSC1 当前值等于预设值中断	13
		14	HSC1 输入方向改变中断	14
		15	HSC1 外部复位中断	15
		16	HSC2 当前值等于预设值中断	16
		17	HSC2 输入方向改变中断	17
		18	HSC2 外部复位中断	18
		32	HSC3 当前值等于预设值中断	19
		29	HSC4 当前值等于预设值中断	20
		30	HSC4 输入方向改变中断	21
		31	HSC4 外部复位中断	22
		33	HSC5 当前值等于预设值中断	23
时基中断	定时	10	定时中断 0	0
		11	定时中断 1	1
	定时器	21	T32 当前值等于预设值中断	2
		22	T96 当前值等于预设值中断	3

中断指令主要有以下几条指令：

中断允许指令（ENI）：全局性地允许所有被连接的中断事件。

中断禁止指令（DISI）：全局性地禁止处理所有中断事件。

中断程序有条件返回指令（CRETI）：在控制它的逻辑条件满足时，从中断程序返回。编

程软件会自动地为各中断程序添加无条件返回。

中断连接指令（ATCH）：用来建立中断事件（EVNT）和处理此事件的中断程序（INT）之间的联系。中断事件由中断事件号指定（见表 3-2），中断程序由中断程序号指定。为某个中断事件指定中断程序后，该中断事件被自动地允许。

中断分离指令（DTCH）：用来断开事件（EVNT）和中断程序（INT）之间的联系，从而禁止单个中断事件。

如图 3-20 所示，这是一个采样模拟输入信号的程序。当 I0.0 为 ON 时，首先给 SMB34（特殊寄存器，用于存储定时中断的时间间隔）写入定时间隔 100ms，连接 10 号定时中断执行 10 号中断程序，并允许全局中断。中断程序 10 执行的功能是读入模拟量通道 AIW0 的数据，并将其存放在 VW20 变量中。EN 为使能输入端，ENO 为使能输出端，当 EN 有效时，指令才被执行，当指令执行后 ENO 为 1，“→|”表示一个可使用的能流，可用可不用。

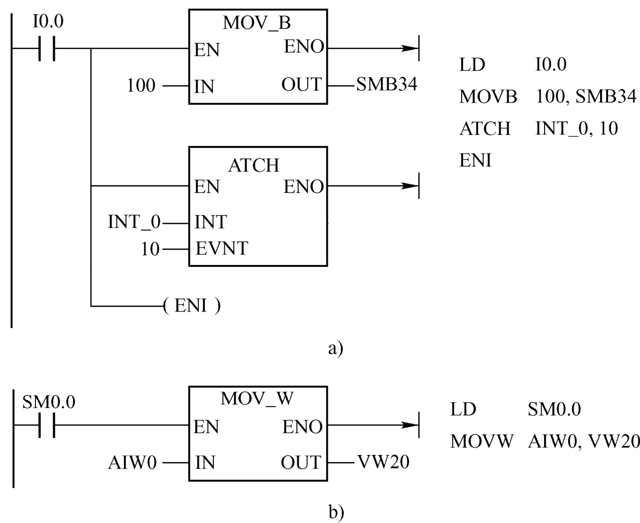


图 3-20 中断指令

a) 主程序 b) 中断程序

3.5 数据处理指令

1. 比较指令

用来比较两个数 IN1 与 IN2 的大小。在梯形图中，满足比较关系式给出的条件时，触点接通。比较关系有等于（=）、大于等于（>=）、小于等于（<=）、大于（>）、小于（<）、不等于（<>）。触点中间的 B、I、D、R、S 分别表示字节比较、整数比较、双整数比较、实数比较和字符串比较。

如图 3-21 所示，在整数比较指令中，当 VW0 >= VW2 并且 VW6<1000 时，Q0.0 为 1；在实数比较指令中，当 VD0 = VD4 或 VD8 < 9.6 时，Q0.1 为 1。

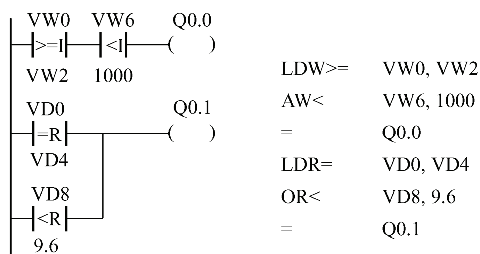


图 3-21　比较指令

2．传送指令

传送指令用于在各个编程元件之间进行数据传送。根据每次传送数据的数量，可分为数据传送指令和数据块传送指令。

数据传送指令每次传送一个数据，MOVB、BIR、BIW、MOVW、MOVD、MOVR 分别表示字节传送、立即读字节、立即写字节、字传送、双字传送和实数传送。数据块传送指令的功能是完成多个数据的传送，BMB、BMW、BMD 分别表示字节块传送、字块传送和双字块传送。如图 3-22 所示，当 I0.0 为 ON 时，执行字传送指令，将 VW10 中的数据传送到 VW20 中，同时利用 ENO 使 Q0.0 为 ON；接着执行立即读指令，将输入 I1 口的状态读入 VB30 中；当 I0.1 为 ON 时，执行字节块的传送指令，将 VB40 开始的 5 个连续的字节数据传送到 VB50 开始的 5 个字节存储单元中，并且使 Q0.1 为 ON。

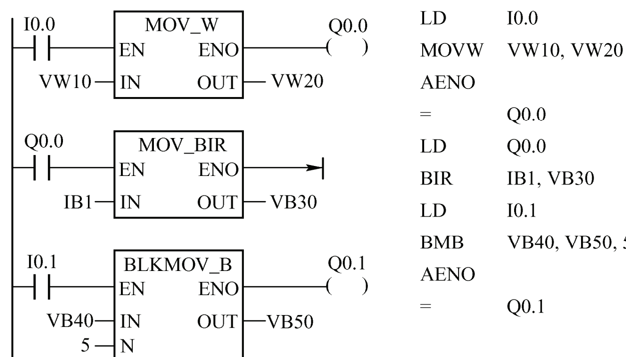


图 3-22　传送指令

【例 3-5】 以 10s 为一个周期，依次循环点亮三盏信号灯；按下启动按钮 I0.0，Q0.0 点亮 3s→Q0.1 点亮 4s→Q0.2 点亮 3s→Q0.0 再次点亮…，不断循环；按下停止按钮 I0.1，信号灯熄灭。

设计要点：1）考虑采用比较指令进行三段输出的切换；2）由于每 10s 循环一次，因此考虑每 10s 定时器复位并重新计时；3）考虑初始化问题，以便每次重新启动，程序按照预定的顺序执行，变量按照设定的值动作。

程序设计如图 3-23 所示。

3．移位指令与循环指令

移位指令将输入 IN 中的数的各位向右或向左移动 N 位后送给输出 OUT。移位指令对移出的位自动补 0，如果移动的位数 N 大于允许值（字节操作数为 8，字操作数为 16，双字操

作数为 32），则实际移位的位数为最大允许值，N 为字节变量。当移位操作结果为 0 时，SM1.0 被置 1；当移位次数大于 0 时，SM1.1 存放最后一次被移出的位的值。SLB、SLW、SLD 分别表示字节左移、字左移和双字左移；SRB、SRW、SRD 分别表示字节右移、字右移和双字右移。

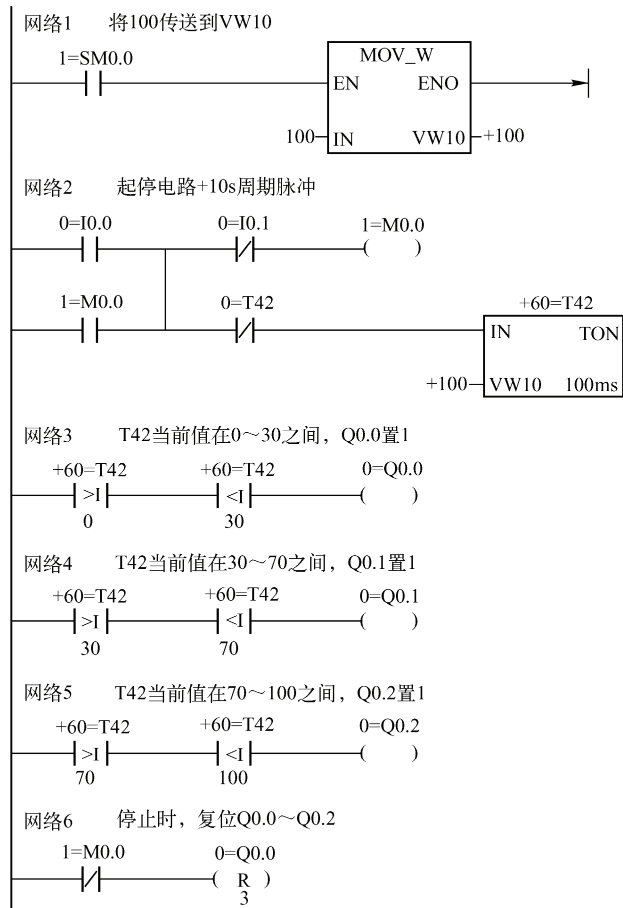


图 3-23　例 3-5 程序

循环移位指令将输入 IN 中的各位向右或向左移动 N 位后，送给 OUT。循环移位是环形的，即被移出来的位将返回到另一端空出来的位置。如果移动的位数 N 大于允许值（字节操作数为 8，字操作数为 16，双字操作数为 32），则执行移位的次数为 N 除以实际数据长度的余数，N 为字节变量，SM1.0 被置 1；当移位次数大于 0 时，SM1.1 存放最后一次被移出的位的值。RLB、RLW、RLD 分别表示字节循环左移、字循环左移和双字循环左移；RRB、RRW、RRD 分别表示字节循环右移、字循环右移和双字循环右移。

如图 3-24 所示。I0.0 为 ON 时，执行一次左移位指令，如果 VB10 移位前的数据是 1000 1100，则左移 2 位后变为 0011 0000，SM1.1 为 0；I0.1 为 ON 时，执行一次循环右移指令，如果 VB50 移位前的数据为 0111 1011，则循环右移 5 位后变为 1101 1011，SM1.1 为 1。

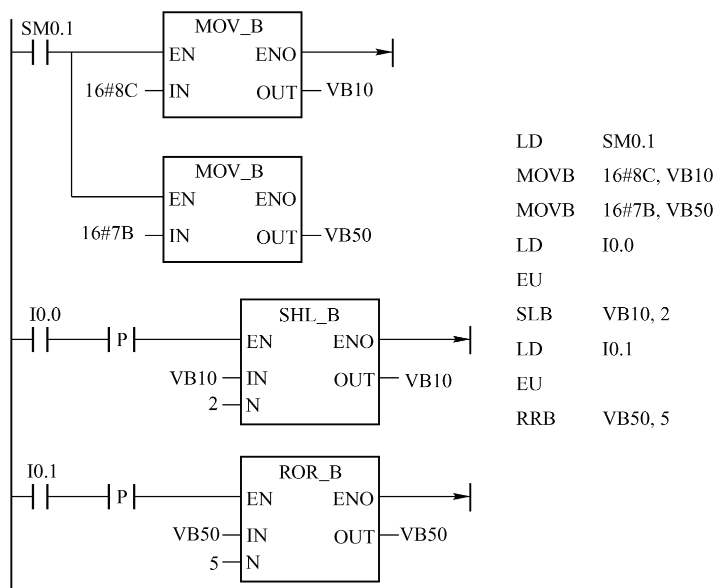


图 3-24　移位指令和循环指令

4．转换指令

（1）数据转换指令

数据转换指令功能是当使能端有效时，将数据 IN 端的数据类型按指令要求转换成相应的数据类型存放到 OUT 中。数据转换指令包括字节（B）与整数（I）之间、整数与双整数（DI）之间、BCD 码与整数之间以及双整数与实数（R）之间的转换。当输出数据超出相应的数据类型范围时，错误标志寄存器 SM1.1 置 1。

如图 3-25 所示，当 I0.1 为 ON 时，首先将 16 位模拟量输入（AIW0）的整数数据转化成双整数数据存放在 AC0 中，再将 VW0 中的整数转换成 BCD 码存放在 VW2 中，然后将 AC0 中的双整数转换成实数。

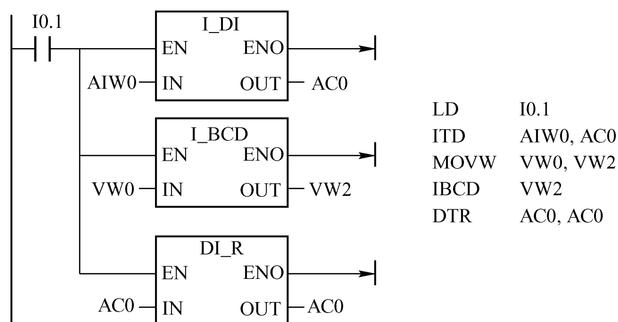


图 3-25　数据转换指令

（2）ASCII 码转换指令

ASCII 码转换指令功能是当使能端有效时，将数据 IN 端的数据类型按指令要求转换成相应的数据类型存放到 OUT 中。ASCII 码转换指令包括 ASCII 码与十六进制数、整数、双整数和实数之间的转换。可进行转换的 ASCII 码为 30~39 和 41~46，对应的十六进制数为 0~9 和

A~F。ATH、HTA、ITA、DTA、RTA 分别表示将 ASCII 码转换成十六进制数、将十六进制数转换成 ASCII 码、整数转换成 ASCII 码、双整数转换成 ASCII 码以及实数转换成 ASCII 码。如图 3-26 所示，当 PLC 上电时，从 VB100 开始的连续 2 个单元的十六进制数（24，56）被转换成 ASCII 码，依次存放到以 VB200 开始的连续的 4 个单元中。

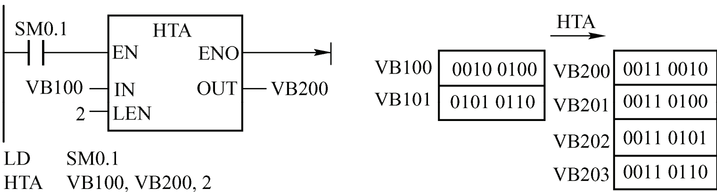


图 3-26 数据转换指令

5．数学运算指令

（1）加减乘除指令

该指令的功能是对有符号整数或实数进行加减乘除运算。加减乘除运算指令包括整数、双整数和实数运算，运算结果分别是整数、双整数和实数。如图 3-27 所示，当 I0.0 为 ON 时，加法指令盒接通，VW10 和 VW12 中的整数进行相加，结果送入 VW12 中；当 I0.1 为 ON 时，执行 MUL_DI 指令，VD20 和 VD24 中的双整数相乘，结果送入 VD24 中。

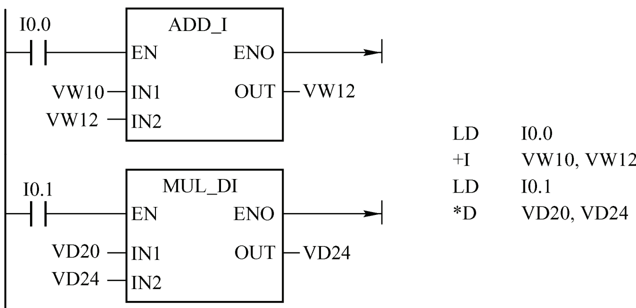


图 3-27 加减乘除指令

（2）加 1 减 1 指令

加 1 减 1 指令就是对有符号数或无符号数进行加 1 或减 1 的操作。加 1 指令包括字节递增 INCB、字递增 INCW 和双字递增 INCD；减 1 指令包括字节递减 DECB、字递减 DECW 和双字递减 DECD。如图 3-28 所示，当 I0.0 的状态从 0 跳变为 1 时，VB0 中的数据就进行自加 1，结果放回 VB0 中；VW2 中的数据进行自减 1，结果放回 VW2 中。

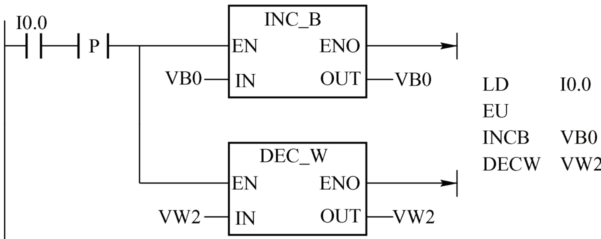


图 3-28 加 1 减 1 指令

（3）逻辑运算指令

逻辑运算指令操作数包括字节、字和双字，该指令的功能是对无符号数进行处理，指令包括逻辑与、逻辑或、逻辑异或和逻辑取反。逻辑与包括字节的逻辑与（ANDB）、字的逻辑与（ANDW）和双字的逻辑与（ANDD）；逻辑或包括字节的逻辑或（ORB）、字的逻辑或（ORW）和双字的逻辑或（ORD）；逻辑异或包括字节的逻辑异或（XORB）、字的逻辑异或（XORW）和双字的逻辑异或（XORD）；逻辑取反包括字节的逻辑取反（INVB）、字的逻辑取反（INVW）和双字的逻辑取反（INVD）。如图 3-29 所示，当 I0.0 为 ON 时，对 VB0 中的数据逻辑取反后存放在 VB2 中；当 I0.1 为 ON 时，VW4 和 VW6 中的数据相与后结果存放在 VW8 中。

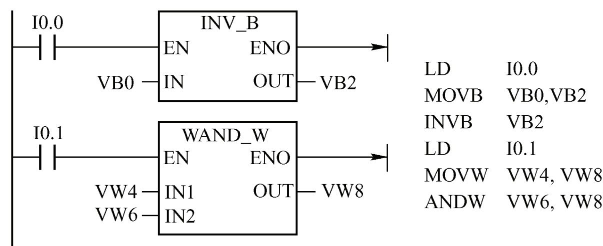


图 3-29　逻辑运算指令

【例 3-6】 分析执行图 3-30 程序后的运行结果。

分析：VB200 中被赋值 16#03，该值为 VW200 的高 8 位，即（VW200）=16#0300，如果连接在输入端 I0.2 的开关开合了一次，则通过执行 ROL_W 指令后，（VW200）=16#0C00；如果连接在输入端 I0.2 的开关开合了两次，则（VW200）=16#3000。

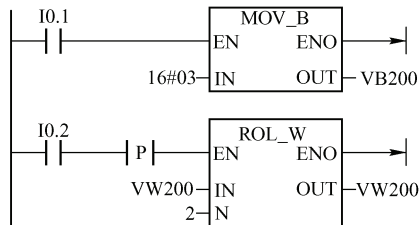


图 3-30　循环移位练习

【例 3-7】 如何在 S7-200 的变量 VB0 中生成全 0 的状态位 00000000；在变量 VB1 中生成全 1 的状态位 11111111。

分析：若两个字节状态相反，则相“与”后可以得到每一位恒定为“0”的变量字节，相“或”后可以得到每一位恒定为“1”的变量字节，程序如图 3-31 所示。

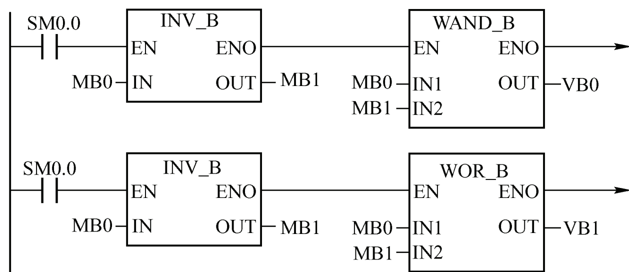


图 3-31　例 3-7 图

程序中，MB0、MB1 为两个标志字节，MB1 为 MB0 的取“反”状态。VB0 为全 0 字节，VB1 为全 1 字节。

3.6　梯形图编程规则

PLC 的几种编程方法中，梯形图编程是使用最多的一种图形编程语言；这是由于梯形图类似于继电器控制系统中的电路图，结构清晰，直观易懂，容易上手，且特别适合开关量逻辑控制的设计。因此在程序设计和编制过程中，须认真掌握梯形图的特点、编程方法和相关规则。

1．梯形图的特点

1）梯形图是一种图形语言，其沿用传统电气图中的继电器触点、线圈、串并联等术语和一些图形符号，左右的竖线称为左右母线。

2）梯形图中接点（触点）只有常开和常闭，接点可以是 PLC 输入点接的开关信号，也可以是 PLC 内部继电器的接点或内部寄存器、计数器等的状态，并且各软元件的动合、动闭触点在编程时都可以无限次使用。

3）梯形图按照从左到右、自上而下的顺序排列，每一行程序都是从左边母线开始，终止于右母线。

4）输入继电器的状态只取决于外部输入信号，梯形图中不能出现输入继电器的线圈；外部负载则只能由输出继电器来驱动。

5）梯形图表示的并不是一个实际电路，而是一个控制程序；触点间的连线表达的是逻辑关系，输出元件也并非真实的继电器线圈等负载元件，而是内部的存储单元，或称为软继电器。程序的运行是一个逻辑运算过程，每条支路的线圈状态可通过支路上的触点状态和连接关系表达出来。

2．梯形图的编程规则与技巧

1）串联电路并联时，应将触点较多的回路放在梯形图最上面。这样可以优化触点结构，并简化程序。如图 3-32 所示，其中图 3-32b 较图 3-32a 更合理。

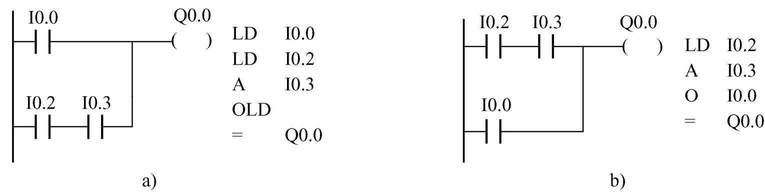


图 3-32　串联触点多的回路放在上面

2）并联电路串联时，应将触点较多的并联回路放在梯形图的最左边。如图 3-33 所示，其中图 3-33b 比图 3-33a 的程序步更少。

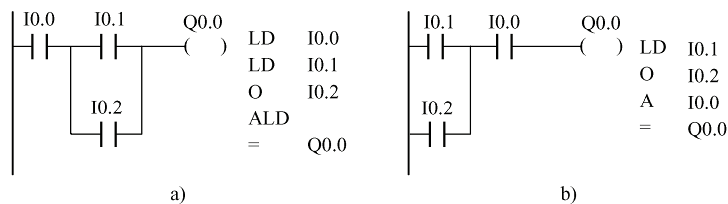


图 3-33　并联触点多的回路放在左边

3）并联线圈回路，从分支点到线圈之间无触点的，线圈应放在上方。如图 3-34 所示，其中图 3-34b 可以不使用 LPS 和 LPP 指令，这就节省了存储器空间，同时缩短了运算周期。

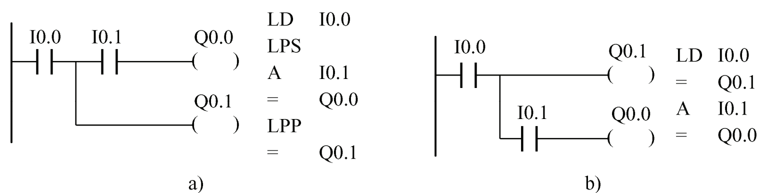


图 3-34　无触点的线圈应放在上方

4）梯形图必须符合顺序执行的原则，即从左到右、从上到下地执行。触点只能水平布置，能流只能单向流动；不符合顺序执行的电路不能直接编程。如图 3-35 所示，其中图 3-35a 所示的桥式回路不能直接编程，必须等效为图 3-35b 的形式方可编程。等效的原则是逻辑关系不变。

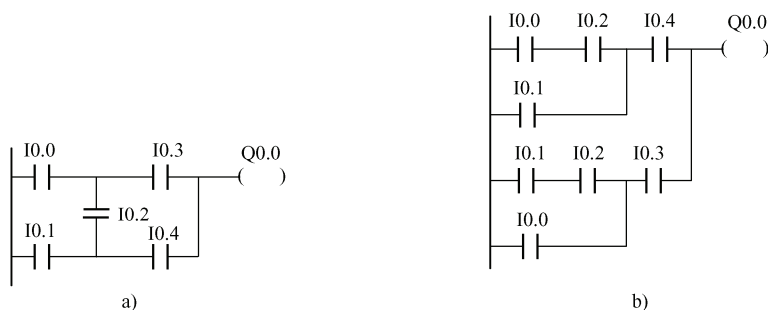


图 3-35　桥式电路的编程

5）避免双线圈输出。同一编号的线圈在一个程序中使用两次称为双线圈输出，双线圈输出容易引起误操作，编程时应避免线圈重复使用。

如图 3-36a 所示，Q0.1 为双线圈输出；在第一行中，当 I0.0=1 时，Q0.1=1；在第二行中，由于 Q0.1 得电，其动合触点闭合，故 Q0.2=1；在第三行中，I0.2=0，将 Q0.1 状态改写为 0，所以经过一个扫描周期后输出为 Q0.1=0、Q0.2=1。在同一个扫描周期中，两个线圈的逻辑运算结果刚好相反，即 Q0.1 的线圈一个“通电”、一个“断电”，由于程序执行完后才将 Q0.1 的状态送到输出模块，所以对于 Q0.1 控制的外部负载来说，真正起作用的是最后一个 Q0.1 的状态。实际编程可以将图 3-36a 改成图 3-36b 的形式。

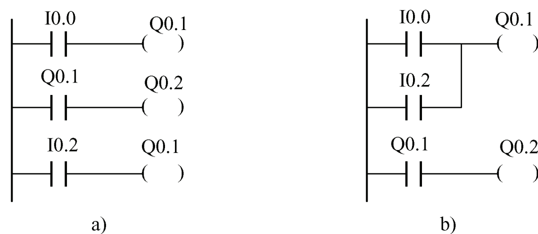


图 3-36　避免双线圈输出

6）输入信号中常闭信号的处理。继电器-接触器控制电路中，从安全角度考虑，一般将

设备由静止状态转为运动状态时（如启动按钮），常连接动合触点；将设备由运动状态转为静止状态时（如停止按钮），常连接动断触点。PLC 系统设计时，同样如此，使用时尽量将停止按钮、热继电器触点等采用动断触点形式接至 PLC，在程序设计时需要注意。

3.7 S7-200 PLC 的基本编程环节

在使用 PLC 基本指令进行编程的过程中，会用到一些常用电路，也称为基本编程环节；熟悉并掌握这些常用编程电路，将会极大地提高编程效率。

3.7.1 用标准触点指令实现一个脉冲输出电路

在 S7-200 PLC 中，要实现一个脉冲宽度为一个扫描周期的脉冲，可以应用标准触点指令实现。如图 3-37 所示，当 I0.0 闭合时，在第一个扫描周期 M0.0 为 ON，而在第二个扫描周期由于 M0.1 为 ON，所以 M0.0 变为 OFF，以后 M0.0 都为 OFF，因此 Q0.0 的输出为一个脉冲输出，脉冲宽度为一个扫描周期。

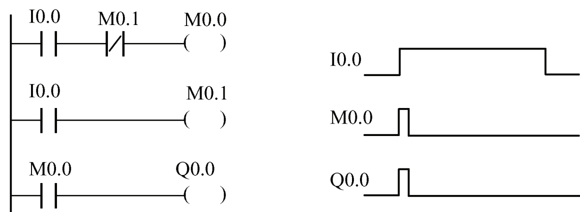


图 3-37 脉冲输出电路

3.7.2 用一个按钮产生启动和停止信号电路

在 PLC 的编程中，可能在硬件上因为输入点数的限制，需要用一个输入点来实现启动和停止两个信号。如图 3-38 所示，当按钮 I0.0 第一次按下时，Q0.0 为 ON，此时 I0.0 为启动信号；当按钮第二次按下时，Q0.0 为 OFF，此时 I0.0 为停止信号。

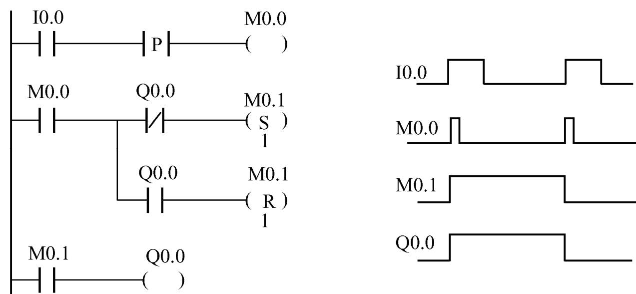


图 3-38 一个按钮产生启动和停止信号电路

练习：也可以采用计数器实现上述要求和功能，试编写相应的梯形图。

3.7.3 分频电路

在许多控制场合，需要对控制信号进行分频。以二分频为例，要求输出脉冲是输入信号

脉冲的二分频。如图 3-39 所示，当 I0.0 由 OFF 变为 ON 时，M0.0 产生一个脉冲，在扫描程序至第三行时，由于 Q0.0 为 OFF，因此 M0.2 不会得电，扫描程序至第四行时，Q0.0 为 ON 并自锁。此后的多个扫描周期中，由于 M0.0 只导通一个扫描周期，因此 M0.2 不会为 ON。当 I0.0 再次由 OFF 变为 ON 时，M0.0 再次产生一个脉冲，此时，因为之前 Q0.0 为 ON，所以 M0.2 也变为 ON，Q0.0 变为 OFF。此后的多个扫描周期中，由于 M0.0 只导通一个扫描周期，Q0.0 一直为 OFF。到下一次 I0.0 由 OFF 变为 ON 时，Q0.0 再为 ON，如此循环，得到的输出 Q0.0 刚好是输入信号 I0.0 的二分频。

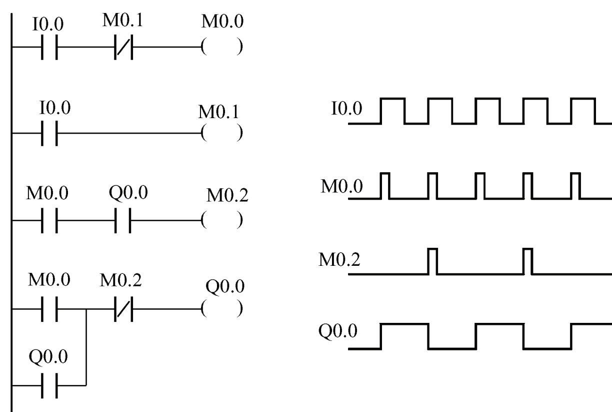


图 3-39　二分频电路

3.7.4　单稳态电路

单稳态电路只有一个稳态。如图 3-40 所示是上升沿触发的单稳态电路，在 I0.0 从 OFF 变为 ON 的上升沿开始，Q0.0 输出一个宽度为 3s 的脉冲，3s 后，T37 动作，其动断触点断开，Q0.0 变为 OFF。I0.0 为 ON 的时间可以大于 3s，也可以小于 3s。

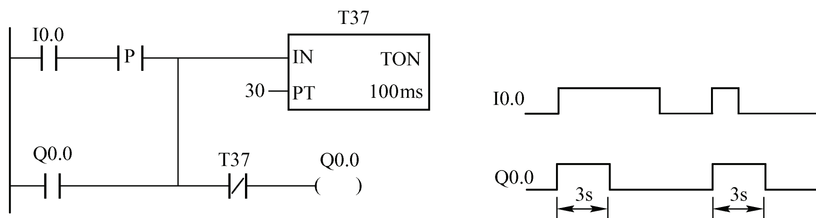


图 3-40　单稳态电路

3.7.5　定时范围的扩展电路

PLC 的定时器的最长定时时间为 3276.7s（16 位数据，分辨率 100ms），如果要得到更长时间的延时，则可以采用如下的方法来实现。

1．定时器接力

采用定时器接力的方式可以实现定时器范围的扩展。所谓定时器接力即先启动一个定时器定时，定时时间到，用第一个定时器的动合触点启动第二个定时器，第二个定时器定时时

间到后，再用第二个定时器的动合触点启动第三个定时器，依次类推，直到所有定时器的设定值之和等于系统要求的定时时间。设各个定时器的设定值分别为 Kt1、Kt2、Kt3、…、Ktn，则对于 100ms 定时器，总的设定时间为 T=0.1×(Kt1+Kt2+Kt3+…+ Ktn)。

如图 3-41 所示，当 I0.0 为 ON 时，定时器 T37 开始定时，定时 3200s 后，T37 的动合触点为 ON，T38 定时器接着开始定时，2000s 后 T38 的动合触点为 ON，T39 定时器接着开始定时，2000s 后 T39 的动合触点变为 ON，使 Q0.0 变为 ON。从 I0.0 为 ON 开始到 Q0.0 为 ON，这段时间总共是 7200s，实现了总共 2 个小时的延时。

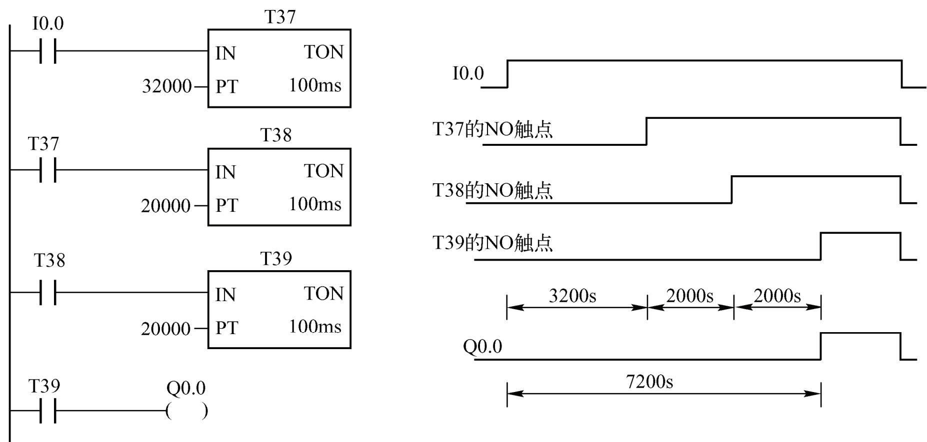


图 3-41　定时器接力定时电路

2．定时器和计数器配合

采用定时器和计数器配合的方式也可以实现定时器范围的扩展。设定时器和计数器的设定值分别是 Kt 和 Kc，则对于 100ms 定时器，总的设定时间为 T=0.1×Kt×Kc。如图 3-42 所示，当 I0.0 为 OFF 时，T39 和 C0 不能工作。当 I0.0 为 ON 时，T39 开始定时，600s 后 T39 的动合触点闭合，动断触点断开。T39 的动断触点的断开导致自身复位，使它自己又重新开始定时，这样定时器 T39 每隔设定的时间（这里为 600s）复位一次。T39 的动合触点每 600s 接通一个扫描周期，使计数器 C0 计一个数，当计到 C0 的设定值（这里为 12）时，C0 的动合触点闭合，Q0.0 变为 ON。从 I0.0 接通到 Q0.0 为 ON，这段时间总共是 0.1×6000×12=7200s，从而实现了 2 个小时的延时。

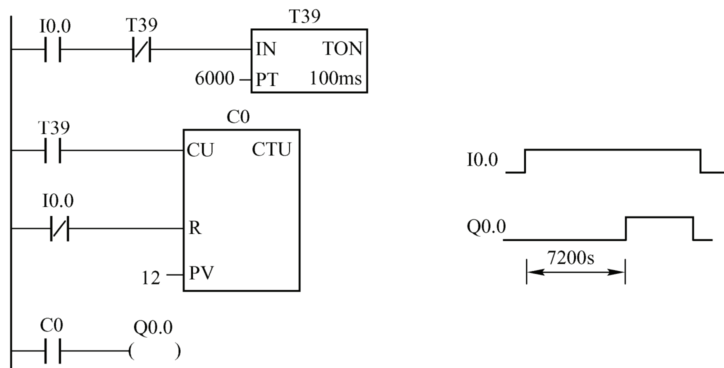


图 3-42　定时器和计数器配合定时电路

3.7.6　闪烁电路

1．使用特殊辅助寄存器

S7-200 系列 PLC 的特殊辅助继电器 SM0.4、SM0.5 分别是 1min、1s 的时钟脉冲，利用这些特殊辅助寄存器可以实现闪烁电路。如图 3–43 所示，当 I0.0 为 ON 时，Q0.1 输出周期为 1s 的脉冲。

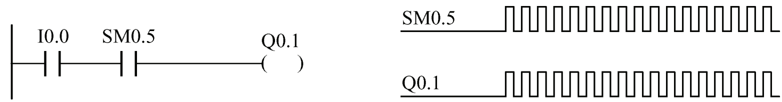


图 3–43　使用特殊辅助继电器的闪烁电路

2．使用定时器

使用特殊辅助继电器实现闪烁电路的缺点是无法输出可调整的脉冲周期和脉冲宽度。利用定时器可以克服这一缺点。如图 3–44 所示，当 I0.1 为 ON 时，定时器 T34 开始定时，2s 后 T34 动合触点接通，Q0.0 变为 ON，同时定时器 T33 开始定时，3s 后 T33 的动断触点断开，T34 被复位，T33 也被复位，Q0.0 变为 OFF，同时 T33 的动断触点又闭合，T34 又开始定时，如此重复。通过调整 T33 和 T34 定时的时间，可以改变 Q0.0 输出 ON 和 OFF 的时间，以此来调整脉冲输出的宽度和周期。

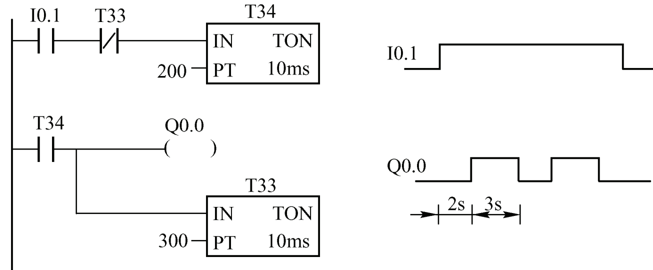


图 3–44　使用定时器的闪烁电路

3.7.7　时序控制电路

可以用多个定时器接力方式来控制时序控制电路中输出继电器的工作。按下启动按钮 I0.0 后，要求 Q0.0 和 Q0.1 按图 3–45 的波形图工作，此时 M1.0 为 ON，T37 定时器开始定时，5s 后 T37 的动合触点闭合，使 T38 定时器开始定时，6s 后 T38 的动合触点闭合，使 T39 定时器开始定时，5s 后 T39 的动断触点断开，T37 被复位，T37 的动合触点断开，又使 T38 被复位，T38 的动合触点断开，从而 T39 也被复位，系统回到初始状态。

从波形图可以看出，Q0.0 的波形图和 T37 的波形图相同，所以可用 T37 的动合触点来控制 Q0.0 的线圈。Q0.1 的波形图可以用 T38 动合触点的波形取反后，再与 M1.0 的波形相与而得到，所以 Q0.1 用 M1.0 的动合触点和 T38 的动断触点组成的串联电路来控制。

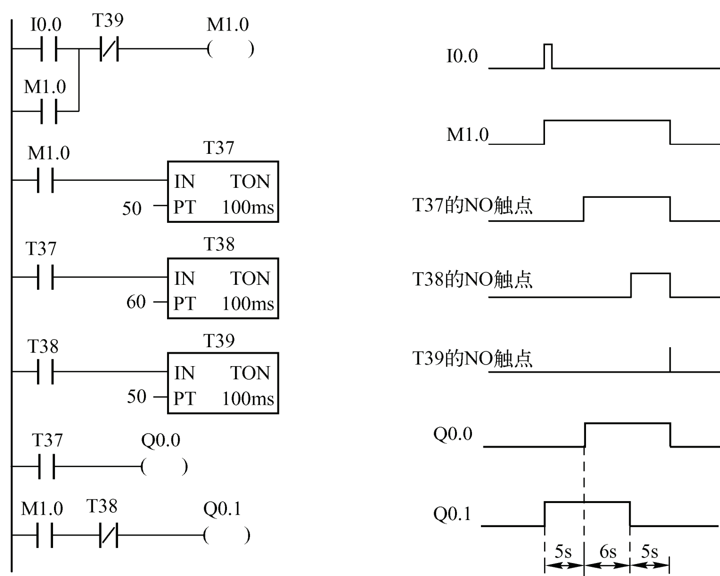


图 3-45　使用多个定时器接力定时的时序电路

第 4 章　S7-200 PLC 程序设计及其应用

4.1　程序设计方法

进行 PLC 控制系统设计时，应根据不同的项目特点选择不同的设计方法，可有效地提高编程效率。常用的 PLC 程序设计方法有继电器控制电路移植法、经验设计法、顺序功能图设计法以及逻辑设计法等。

继电器控制电路移植法常用于对原采用继电器-接触器控制（J-C）的系统进行 PLC 技术改造，这时可根据原有的 J-C 控制系统，将其控制逻辑转换为 PLC 的控制程序来实现，这种设计方法简单实用，调试方便，易于上手，但仅用于对已有老系统的改造。

经验设计法是沿用继电器-接触器控制电路的设计方法来设计梯形图，即在一些典型控制电路的基础上，根据被控对象对控制系统的要求，不断地修改和完善梯形图，直至完全满足各项控制要求；经验设计法一般都需经过多次反复的调试和修改，最终才能得到一个较为满意的结果。这种设计方法没有普遍规律可以遵循，设计所用的时间、质量与设计者的经验有很大关系；它主要用于逻辑关系较为简单的梯形图程序设计。

顺序控制设计法又称为顺序功能图设计法（SFC），它是按照生产工艺预先规定的顺序，在各个输入信号的作用下，根据内部状态和时间的顺序，在生产过程中各个执行机构自动有序地进行操作。功能流程图（简称功能图）又叫状态流程图或状态转移图，它是专用于工业顺序控制程序设计的一种功能说明性语言，能完整地描述控制系统的工作过程、功能和特性，是分析、设计电气控制系统控制程序的重要工具。这种方法能够清晰地表示出控制系统的逻辑关系，从而大大提高编程的效率。

4.2　继电器控制电路移植法

在进行 PLC 控制系统设计时，如果是对原采用继电器-接触器控制的系统进行 PLC 技术改造，或者控制系统可以采用继电器-接触器控制电路来实现；此时可选用移植法来完成电气系统的 PLC 设计。

这是因为原有的继电器-接触器控制系统已经过长期的使用和实践，可以满足系统的全部控制要求，而 PLC 梯形图与继电器控制电路原理图在表达方式和分析方法上都是十分相似的，所以可以根据继电器电路图直接设计 PLC 梯形图，也就是把继电器控制电路直接转化为具有相应功能的 PLC 外部接线图和梯形图。

继电器控制电路移植法是进行 PLC 控制系统设计时，尤其是用 PLC 对继电器控制系统进行技术改造时，常使用的一种简便、可靠的设计方法。但需注意移植法不是简单的代换，设计时必须确保所获得的梯形图与原继电器控制电路图等效。

4.2.1 两台电动机顺序控制的实现

1．任务描述

完成两台电动机的顺序起动、逆序停止的控制。

以锅炉鼓风机和引风机的电气控制为例，锅炉的鼓、引风机的作用是用来保障燃料充分燃烧，并维持锅炉房卫生环境，因此鼓、引风机需配合以确保锅炉炉膛为微负压。

对锅炉鼓、引风机的电气控制提出如下要求：两台电动机起动时，必须先起动引风机再起动鼓风机；停止时，必须在鼓风机停止后，方可手动停止引风机。

2．任务目标

原控制系统已经采用继电器-接触器控制方式实现，现要求采用 PLC 进行技术改造。

采用继电器-接触器控制设计的主电路、控制电路如图 4−1 所示。

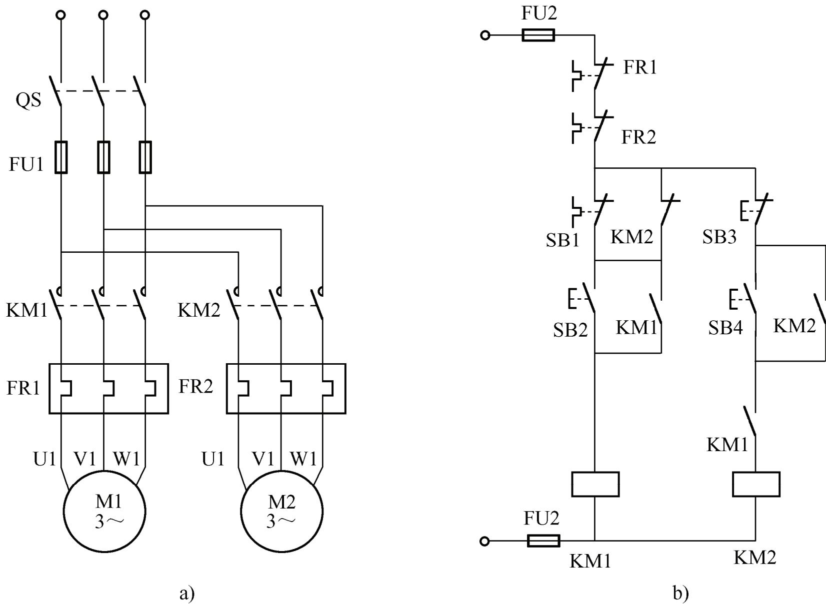


图 4−1 电动机顺序控制电路原理图

a) 主电路 b) 控制电路

在图 4−1a 中，M1 为引风机拖动电动机，由接触器 KM1 控制；M2 为鼓风机拖动电动机，由接触器 KM2 控制；热继电器 FR1、FR2 为两台电动机提供过载保护。

在图 4−1b 中，SB1、SB2 是引风机电动机的停止、起动按钮；SB3、SB4 是鼓风机电动机的停止、起动按钮；两条控制支路都是简单的起保停控制电路，不同之处在于鼓风机控制支路中，串入了引风机控制接触器 KM1 的动合（NO）触点，确保只有引风机电动机启动后，鼓风机才能起动；而引风机控制支路中，在停止按钮两端并联了 KM2 的动断（NC）触点，保证鼓风机停止后，引风机才可手动停止；满足了实际控制要求。

3．知识储备

该任务可采用 PLC 移植法完成。采用移植法进行设计时，原系统主电路保持不变，只需将控制电路替换为具有相应功能的 PLC 外部接线图和梯形图即可。基本步骤如下：

1）熟悉被控设备生产工艺和动作顺序，掌握电气控制系统工作原理。

2）统计系统输入/输出点数，完成 PLC 选型和 I/O 点的分配。

3）绘制 PLC 外部接线图，并将原有控制电路转化为梯形图。

4）程序录入、调试、验证并确保系统正常工作。

4．任务实施

（1）PLC 选型和 I/O 点的分配

随着 PLC 技术的发展，PLC 产品的种类也越来越多。不同型号的 PLC，其结构形式、性能、容量、指令系统、编程方式及价格也各有不同，适用的场合各有侧重。因此，合理选用 PLC，对于提高 PLC 控制系统的技术和经济指标有着重要的意义。

PLC 的选择主要从 PLC 的机型、容量、I/O 模块、电源模块、特殊功能模块以及通信联网能力等方面加以综合考虑。

1）机型的选择：PLC 按结构分为整体式和模块式两类。整体型 PLC 体积小、价格便宜，一般用于系统工艺过程较为固定的小型控制系统中；模块型 PLC 配置灵活，可根据需要选配不同模块组成一个系统，在 I/O 点数等方面选择余地大，而且装配方便，便于扩展和维修，一般用于较复杂的中大型控制系统。

2）输入/输出点数（I/O）的估算：I/O 点数越多，PLC 价格也越高，因此在满足控制要求的前提下应尽量减少 I/O 点数，但 I/O 点数估算时还必须考虑适当的余量，以备今后系统改进或扩展时使用。通常根据统计的输入/输出点数，再增加 10%～20%的裕量。

3）存储器容量的估算：存储器容量是可编程序控制器本身能提供的硬件存储单元的大小，程序容量是存储器中用户应用程序使用的存储单元的大小，因此程序容量小于存储器容量。设计阶段，由于用户应用程序还未编制，因此，程序容量在设计阶段是未知的，需在程序调试之后才知道。为了设计选型时能对程序容量有一定估算，通常采用存储器容量的估算来替代。

存储器容量的估算没有固定的公式，许多文献资料中给出了不同公式，如果控制系统只有开关量控制时，大体上都是按数字量 I/O 点数的 10～15 倍来计算开关量控制时所需的存储器字数，然后再考虑 20％～30％的裕量。

4）控制功能的选择：该选择包括运算功能、控制功能、通信功能、编程功能、诊断功能和处理速度等特性的选择。

根据以上 PLC 选型原则及对控制系统的分析，这个控制系统的输入信号有热继电器 FR1、FR2 触点，引风机电动机的停止、起动按钮 SB1、SB2，鼓风机电动机的停止、起动按钮 SB3、SB4，共 6 个开关量信号；为节省输入点数，实际接线时，在满足控制要求前提下，将热继电器 FR1、FR2 的两个动断触点串联后接入 PLC，可节省一个输入点，共计 5 点输入；输出有驱动鼓、引风机电动机工作的交流接触器线圈，共 2 个输出信号。根据统计的 I/O 点数及预完成的系统功能，可选择 S7-200 整体式 PLC CPU221，该 PLC 有 6 个输入点、4 个输出点，类型 AC/DC/Relay，能满足系统控制要求。

PLC 的 I/O 点的地址分配见表 4-1。

表 4-1　PLC 的 I/O 地址分配表

PLC 的 I/O 地址	连接的外部设备	在控制系统中的作用
I0.0	SB1（NC）	引风电动机停止
I0.1	SB2（NO）	引风电动机起动
I0.2	SB3（NC）	鼓风电动机停止
I0.3	SB4（NO）	鼓风电动机起动
I0.4	FR1（NC）、FR2（NC）串联	电动机过载保护
Q0.0	正转接触器线圈	控制电动机正转
Q0.1	反转接触器线圈	控制电动机反转

注：按照控制要求，两台电动机中任何一台过载时，系统都会停止运行，所以设计外部接线时，可将 FR1、FR2 两个动断输入信号串接后接入 PLC，这种接法可节省输入点。

还需注意的是，在 I/O 地址分配时，从安全角度考虑，停止按钮、热继电器控制触点等应该使用动断触点接入。这是因为这些安全系数要求较高的触点，如果选择动合触点连接，会影响到系统的安全运行，比如停止按钮若选择动合触点接入，当连接线因意外断开时，系统将无法正常停止。

（2）PLC I/O 外部接线图

根据表 4-1，将 PLC 与外部设备连接起来，接线图如图 4-2 所示。

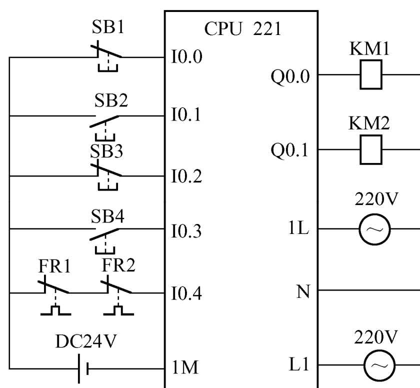


图 4-2　电动机顺序控制 PLC I/O 外部接线图

（3）程序的实现

采用移植法设计程序时，只需根据 I/O 分配情况将控制电路替换为梯形图即可。替换后的梯形图程序如图 4-3a 所示；为了优化程序，减少程序步数，可将图 4-3a 调整为图 4-3b 的形式。

4.2.2　星形-三角形降压起动控制的实现

1．任务描述

完成异步电动机星形-三角形（Y-≥）降压起动的控制。

三相异步电动机星形-三角形（Y-≥）降压起动方式，企业通常采用继电器-接触器构成的 Y-≥启动柜来实现。

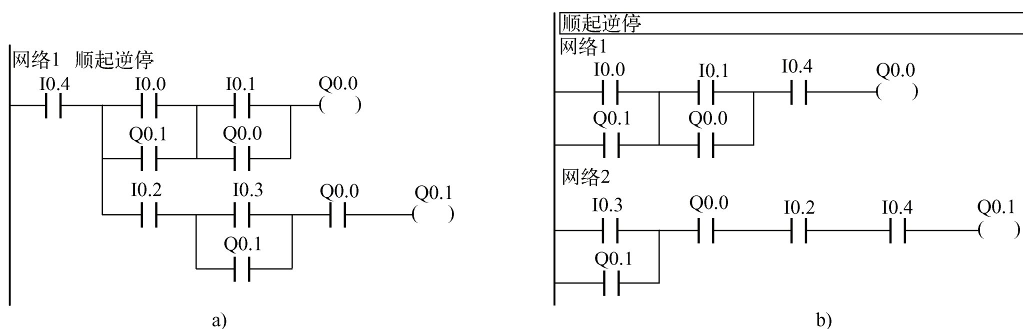


图 4-3　电动机顺序控制梯形图

对于正常运行时定子绕组接成三角形的较大容量的三相笼型异步电动机，可采用星形-三角形降压起动方法达到限制起动电流的目的。

2．任务目标

原控制系统采用继电器-接触器组成的Y-≥启动柜来实现，现要求采用 PLC 进行技术改造。其主电路、控制电路如图 4-4 所示。

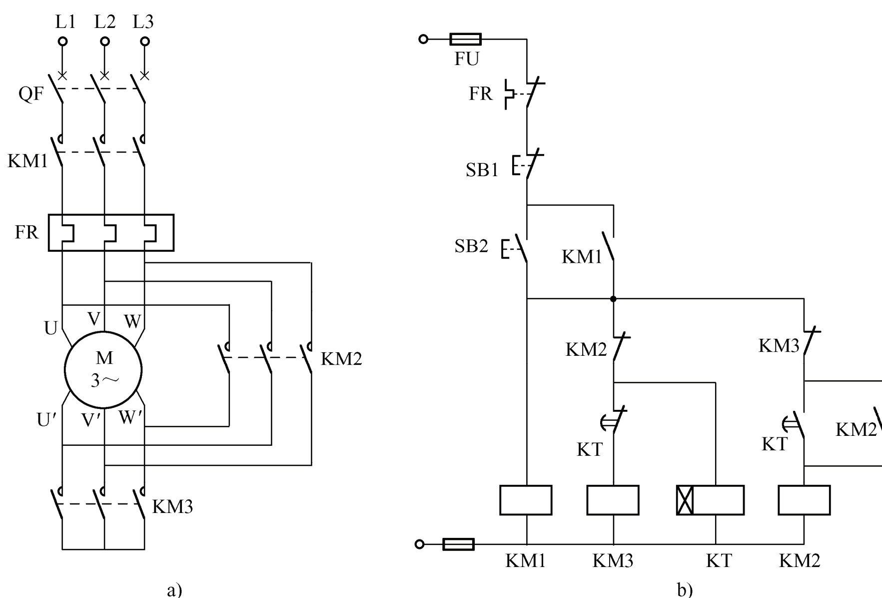


图 4-4　星-三角降压起动控制电路原理图

该电路采用了三个接触器，接触器 KM1、KM3 线圈得电，电动机星形降压运行；接触器 KM1、KM2 线圈得电，电动机三角形全压运行。时间继电器 KT 完成启动时间的设定和状态的切换；SB1 为停止按钮，SB2 为起动按钮。

3．知识储备

该任务同样采用 PLC 移植法完成。采用移植法改造继电器-接触器控制系统时，原系统中的中间继电器、时间继电器以及计数器等硬件设备均可省去不再使用，而采用 PLC 内部对应的软继电器替换；进行系统设计时，应进行软继电器资源分配，确定对应的软继电器编号。

另外在改造时，还需注意以下几点：

1）PLC 采用周期扫描工作方式，与继电器-接触器控制系统是不同的，所以移植法并不是简单的梯形图转换，而应确保替换后的梯形图与原系统控制电路等效。

2）输入触点的状态。PLC 在连接外部主令电器或传感器时，如按钮、热继电器触点、接近开关等，可选择连接常开、常闭两种状态；一般情况下，应尽量连接动合触点；有时从安全角度考虑，需要连接动断触点，这时要注意梯形图设计的逻辑关系。

3）继电器-接触器系统设计时，因受硬件触点数量的限制，电路中许多触点都是多条支路共用的，导致控制电路结构复杂；而 PLC 内部触点可无限使用，因此转换时，可以通过增加 PLC 程序中的“软”触点或辅助继电器等方式简化电路。

4．任务实施

（1）PLC 的选型和 I/O 点的分配

根据电路图，该系统的输入信号有热继电器 FR 触点、停止按钮 SB1 及起动按钮 SB2 共 3 个控制信号；根据控制要求，连接时可将 FR 的 NC 触点与停止按钮 SB1 的 NC 触点串联接入 PLC，可节省一个输入点。输出有 KM1、KM2、KM3 交流接触器线圈，共 3 个输出信号。时间继电器 KT 功能可通过 PLC 内部定时器来实现。

根据 I/O 点数，可选择 S7-200 整体式 PLC CPU221，类型 AC/DC/Relay。PLC 的 I/O 点的地址分配见表 4-2。

表 4-2　PLC 的 I/O 地址分配表

PLC 的 I/O 地址	连接的外部设备	在控制系统中的作用
I0.0	FR1（NC）、SB1（NC）串联	停止及过载保护
I0.1	SB2（NO）	引风电动机起动
Q0.0	KM1 接触器线圈	电源控制
Q0.1	KM2 接触器线圈	电动机三角形接法
Q0.2	KM3 接触器线圈	电动机星形接法

（2）PLC I/O 外部接线图

根据表 4-2，将 PLC 与外部设备连接起来，接线图如图 4-5 所示。注意应在 PLC 外部设置 KM2 和 KM3 的辅助动断触点组成的硬件互锁电路。

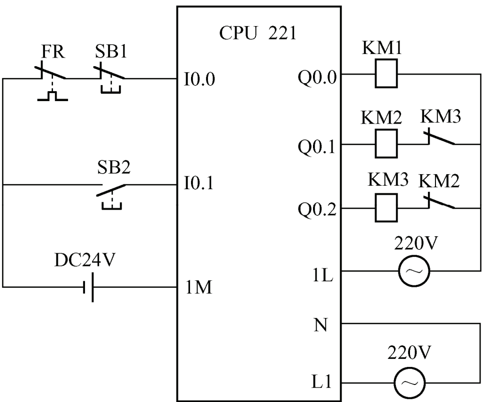


图 4-5　星-三角降压起动控制 PLC 外部接线图

（3）程序的实现

按照继电-接触器控制电路直接转换后的 PLC 梯形图如图 4-6a 所示，图 4-6b 为优化后的程序。

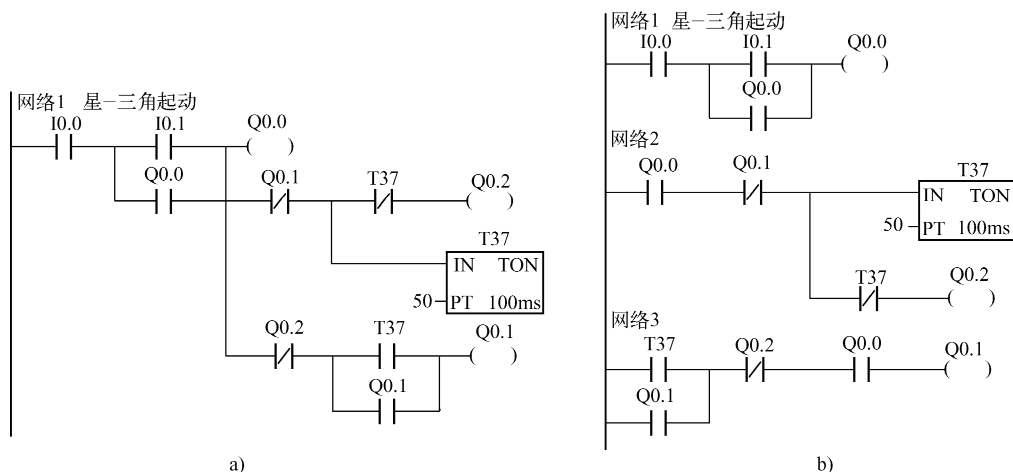


图 4-6 星-三角降压起动控制梯形图

4.3 经验设计法

经验设计法是在一些典型控制电路的基础上完成程序设计，这种方法主要是依靠设计人员的积累和经验进行设计。经验设计法没有规律可循，具有很大的试探性和随意性，往往需经多次反复修改和完善才能符合设计要求，所以设计的结果往往不很规范，因人而异。经验设计法一般适用于设计一些简单的梯形图程序或复杂系统的某一局部程序（如手动程序等）。

用经验设计法设计 PLC 程序时大致步骤如下：第一，分析控制要求、确定控制原则；第二，统计主令电器和检测元器件数量，确定输入输出设备；第三，分配 PLC 的 I/O 点及内部软元件资源；第四，设计执行元件的控制程序；第五，对照控制要求，检查、修改和完善程序。

4.3.1 电动机正反转控制的实现

1．任务描述

完成异步电动机正反转（正-反-停）控制。

2．任务目标

采用 PLC，实现单台电动机正-反-停控制，即：

1）按下正转起动按钮时：

① 若此前电动机为停止状态，则电动机正转起动，并保持正转运行。

② 若此前电动机为反转运行，则将电动机切换到正转状态，并保持电动机正转。

③ 若此前电动机已经是正转，则转动状态不变。

电动机正转状态一直保持到有反转按钮或停止按钮按下为止。

2）按下反转起动按钮时：

① 若此前电动机停止，则电动机反转起动，并保持反转运行。

② 若此前电动机正转，则将电动机切换到反转状态，并保持电动机反转。

③ 若此前电动机的状态已经是反转，则电动机的转动状态不变。

电动机反转状态一直保持到有正转按钮或停止按钮按下为止。

3）按下停止按钮时：

电动机停止运行，系统停止工作。

4）为避免出现电源短路情况，必须进行正反转互锁控制。

3．任务实施

（1）电气主电路的实现

电气主电路如图 4-7 所示，如果控制电路使接触器 KM1 线圈得电，则 KM1 的主触点吸合，电动机正转；如果控制电路使接触器 KM2 线圈得电，则 KM2 的主触点吸合，电动机反转。为避免电动机长期过载损坏，采用热继电器 FR 进行过载保护。

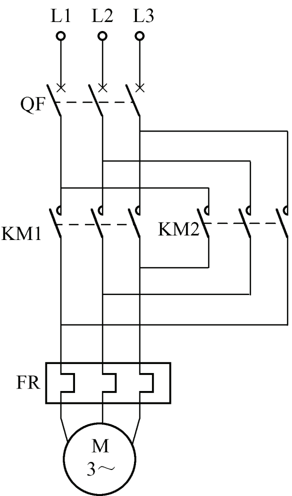


图 4-7　PLC 控制电动机正反转电气主电路图

（2）PLC 的选型和 I/O 点的分配

根据 PLC 选型原则及对电动机控制系统的分析，控制系统的输入有电动机正转起动按钮、电动机反转起动按钮、电动机停止按钮及热继电器触点共 4 个输入点；输出有驱动电动机正反转工作的交流接触器线圈共 2 个负载。故可以选择 S7-200 系列如 CPU221 整体式 PLC，该模块提供 6 点数字量输入，4 点数字量输出，所以不再需要输入/输出模块。PLC 的 I/O 点的地址分配见表 4-3。

表 4-3　PLC 的 I/O 地址分配表

PLC 的 I/O 地址	连接的外部设备	在控制系统中的作用
I0.0	停止按钮（常闭）	系统停止
I0.1	正向起动按钮（常开）	电动机正转命令
I0.2	反向起动按钮（常开）	电动机反转命令

（续）

PLC 的 I/O 地址	连接的外部设备	在控制系统中的作用
I0.3	热继电器触点（常闭）	电动机过载保护
Q0.0	正转接触器线圈	控制电动机正转
Q0.1	反转接触器线圈	控制电动机反转

（3）PLC I/O 外部接线图

根据表 4-3，将 PLC 与外部设备连接起来，接线图如图 4-8 所示。

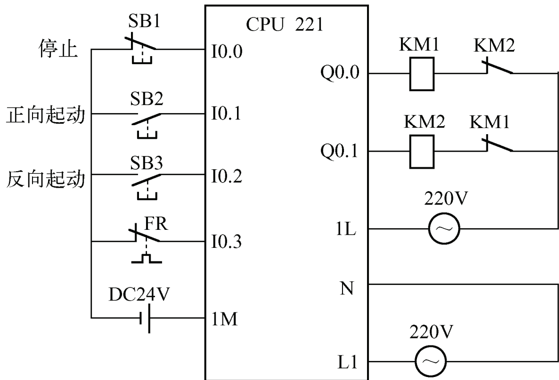


图 4-8　PLC 控制电动机正反转 I/O 外部接线图

由于 KM1 和 KM2 在切换得电过程中存在电感的延时作用，可能会出现一个接触器还未断弧，另外一个却已合上的现象，从而造成瞬间短路故障；或者由于某一接触器的主触点被断电时产生的电弧熔焊而粘结，其线圈断电后主触点仍然是接通的，这时如果另一接触器的线圈通电，仍然会造成三相电源短路事故。为了防止短路故障的出现，在 PLC 外部设置了 KM1 和 KM2 的辅助动断触点组成的硬件互锁电路。

（4）程序的实现

由于经验设计法没有固定的方法和普遍的规律可以遵循，所以设计程序时，应先从简单的典型控制电路入手，逐步添加并实现各项控制功能，设计和完善过程如下：

电动机初步正转控制电路→电动机初步正反转控制电路→电动机正反转的互锁电路→电动机正反转的切换电路→电动机正反转控制的实现。具体过程如图 4-9～图 4-13 所示。

从图 4-9 可以看出，起动按钮 I0.1 持续为 ON 的时间一般都很短，这种信号称为短信号，如何使线圈 Q0.0 保持接通状态呢？可以利用线圈自身的动合触点使线圈保持通电即“ON”状态，这种功能称为自锁或自保持功能。自保持控制电路常用于无机械锁定开关的启停控制。

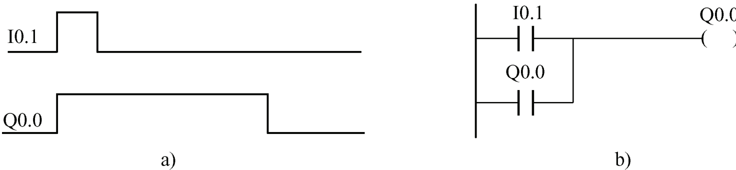


图 4-9　电动机正转控制电路

a) 时序图　b) 梯形图

系统要求电动机不能同时进行正转和反转，如图 4-11 所示在梯形图中将 Q0.0 和 Q0.1 的动断触点分别与对方的线圈串联，可以保证它们不会同时为 ON，因此，KM1 的线圈和 KM2 的线圈不会同时通电。这种安全措施在传统的继电器-接触器控制电路中称为“互锁”。

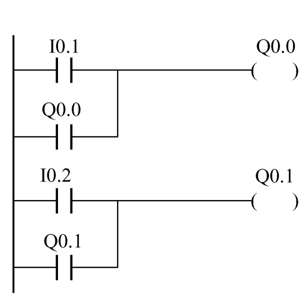


图 4-10　电动机正反转控制电路

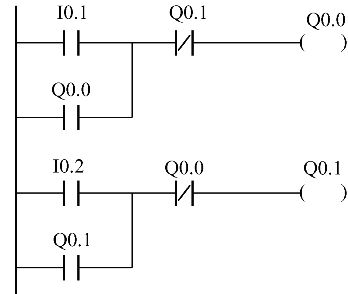


图 4-11　电动机正反转的互锁电路

此外根据控制要求按钮连锁，即利用正转按钮切断反转的控制通路，利用反转按钮来切断正转的控制通路。按钮连锁的梯形图如图 4-12 所示。

当按下停止按钮时，无论在此之前电动机的转动状态如何，都停止电动机的转动。利用停止按钮 I0.0 同时切断正转和反转的控制通路，停止功能的逻辑实现如图 4-13 所示。由于 PLC 输入端子 I0.0 连接的是停止按钮的动断触点，未按压时逻辑状态始终为 1，故软元件 I0.0 动合触点接通，动断触点断开；所以设计时应连接软元件动合触点。

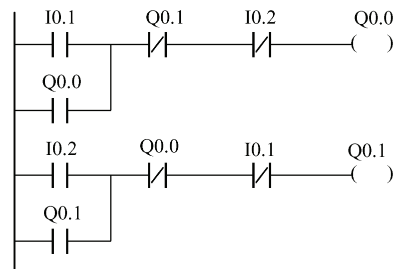


图 4-12　双重连锁的正反转控制电路

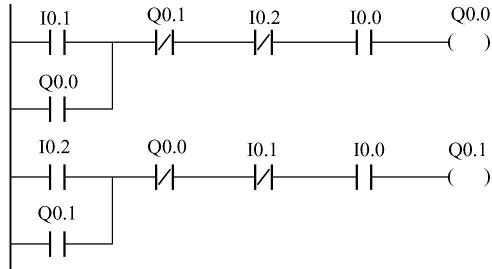


图 4-13　电动机停止控制

考虑电动机的过载保护，本系统采用热继电器的动断触点作为过载时 PLC 的输入控制信号。电动机正常工作时，FR 动断触点持续接通，PLC 的输入继电器 I0.3 得电，I0.3 动合触点动作；当电动机过载状态，热继电器 FR 动作，其动断触点断开，PLC 的输入继电器 I0.3 将失电，将正反转电路切断。图 4-14 梯形图是实现所有控制功能的完整程序。

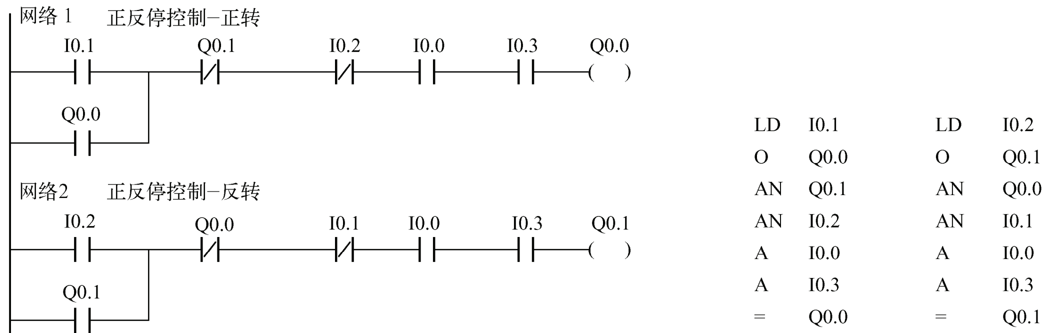


图 4-14　电动机正反转控制电路梯形图及语句表

4.3.2　花样喷泉控制功能的实现

1．任务描述

采用 PLC 完成三组喷头的花样喷泉控制。

2．任务目标

具体的控制要求为：

1）喷泉有 A、B、C 共 3 组喷头，由三台水泵控制。

2）按下开始按钮后，A 组先喷水，10s 后停；然后 B 组和 C 组同时喷水。

3）B 组和 C 组喷水 10s 后 B 组停，再 10s 后 C 组停。

4）A、B 组同时喷水，5s 后，C 组也喷水，持续 10s 后全停。

5）再 5s 后重复 2）～5）步。

6）按下停止按钮后，A、B、C 3 组喷头全部停止。

3．知识储备

在采用 PLC 设计按照时间顺序动作或按照流程顺序动作的系统时，可考虑先画出控制流程图，这将有利于系统的设计和调试。

根据控制要求绘制花样喷泉流程图如图 4-15 所示。

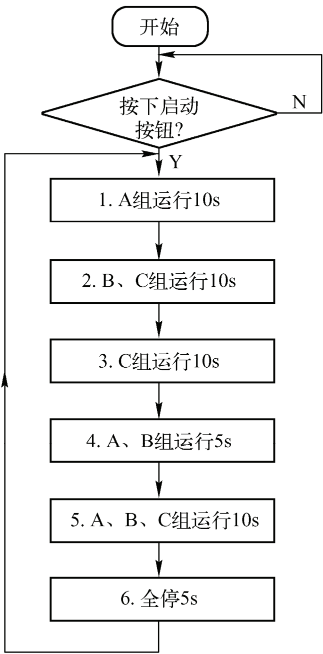


图 4-15　花样喷泉流程图

4．任务实施

（1）PLC 的选型及系统资源的分配

根据控制要求，有启动按钮和停止按钮两个输入点，有 A、B、C 三组喷头共 3 个输出点。可选择 CPU221 PLC，6 点输入/4 点输出，类型 DC/DC/DC，PLC 输入/输出地址分配见表 4-4。

表 4-4　PLC 的 I/O 地址分配表

PLC 的 I/O 地址	连接的外部设备	在控制系统中的作用
I0.0	SB1（常开）	系统开始按钮
I0.1	SB2（常闭）	系统结束按钮
Q0.0	A 组喷头 KM1 线圈	A 组水泵工作
Q0.1	B 组喷头 KM2 线圈	B 组水泵工作
Q0.2	C 组喷头 KM3 线圈	C 组水泵工作

在整个运行周期中，Q0.0、Q0.1、Q0.2 输出触点多次得电失电，为了书写的方便和避免使用双线圈，可使用 PLC 内部辅助继电器 M 作标志位来表示每一阶段工作的状态，PLC 的内部辅助继电器地址分配见表 4-5。采用定时器对流程图中的每一阶段进行计时，PLC 定时器地址分配见表 4-6。

表 4-5　PLC 辅助继电器地址分配表

PLC 的 M 地址	在控制系统中的作用
M0.0	流程图第 1 步辅助继电器
M0.1	流程图第 2 步辅助继电器
M0.2	流程图第 3 步辅助继电器
M0.3	流程图第 4 步辅助继电器
M0.4	流程图第 5 步辅助继电器
M0.5	流程图第 6 步辅助继电器

表 4-6　PLC 定时器地址分配表

PLC 的 T 地址	在控制系统中的作用
T37	A 组喷 10s 计时
T38	B、C 组喷 10s 计时
T39	C 组喷 10s 计时
T40	A、B 组喷 5s 计时
T41	A、B、C 组喷 10s 计时
T42	A、B、C 组停喷 5s 计时

（2）PLC I/O 外部接线图

PLC I/O 外部接线如图 4-16 所示。

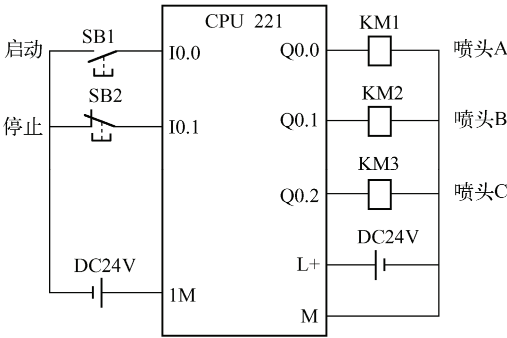


图 4-16　花样喷泉 PLC I/O 外部接线图

（3）程序的实现

1）根据流程图逐步编写梯形图程序，如图 4-17 所示。

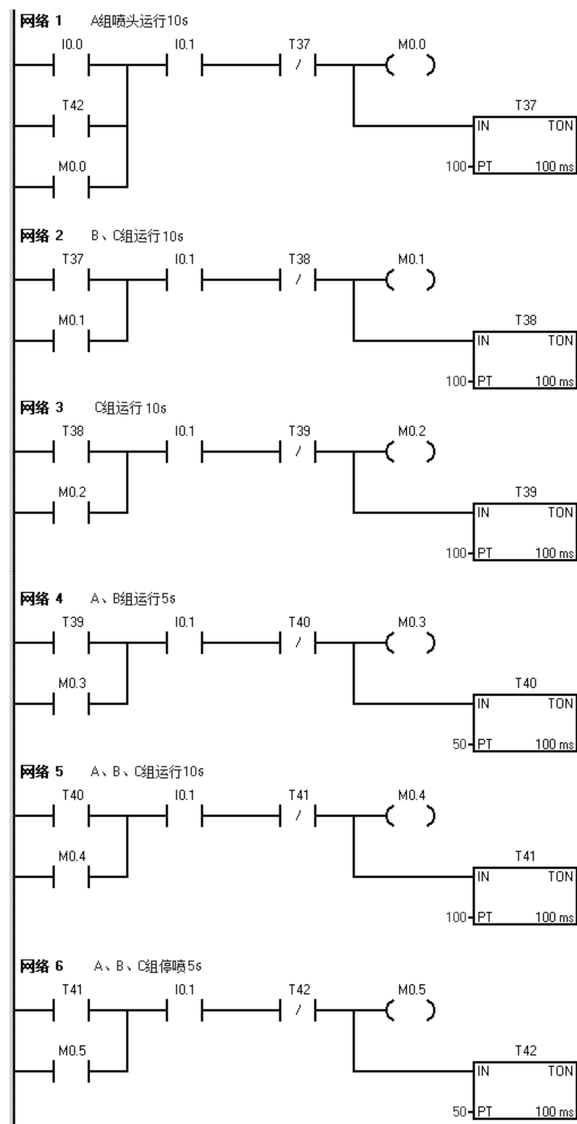


图 4-17 根据流程图编写梯形图

2）三组负载驱动情况梯形图如图 4-18 所示。

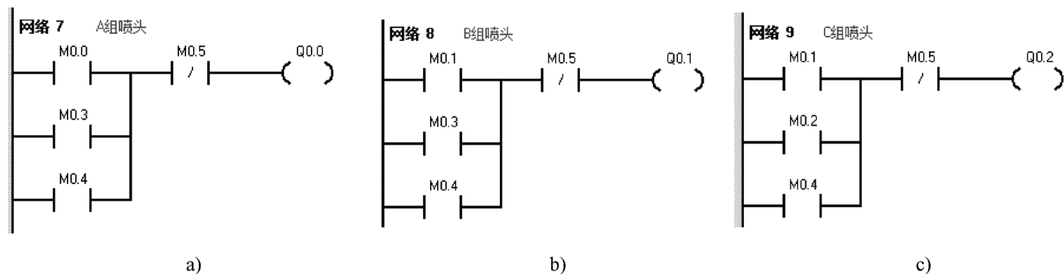


图 4-18 A、B、C 组喷头工作情况

a) A 组喷头 b) B 组喷头 c) C 组喷头

4.4 顺序控制设计法

经验设计法对于一些比较简单的程序设计还是比较奏效的，可以收到快速、简捷的效果。但用经验设计法设计的梯形图，是按照设计者的经验和思维习惯进行设计的，因此没有一套固定的方法和步骤可以遵循，具有很大的试探性和随意性，一个程序往往需要经多次的反复修改和完善才能满足控制要求，所以设计的结果也是因人而异。对一些复杂的、经验法难以奏效的程序设计，就可以考虑采用其他的设计方法。

顺序控制设计法又称为功能图设计法（Sequential Function Chart，SFC），功能图也叫状态流程图或状态转移图，它是专用于工业顺序控制程序设计的一种功能说明性语言，能完整地描述控制系统的工作过程、功能和特性，是分析、设计电气控制系统控制程序的重要工具。

4.4.1 液压动力滑台运动过程的实现

1．任务描述

液压动力滑台是组合机床用来实现进给运动的通用部件，动力滑台通过液压传动可以方便地进行换向和调速的工作，其电气控制系统过去多采用继电器控制，但接线复杂，可靠性低，目前多采用 PLC 控制。

2．任务目标

该任务采用 PLC 完成液压动力滑台在三位置间运动的控制。在实际工作时运动过程是：快进→工进→快退，这三个运动过程由快进、工进和快退三个电磁阀控制。

图 4-19 所示为滑台运动示意图，在原点（SQ1）处，按下启动按钮，滑台按照预定的顺序依次运行。

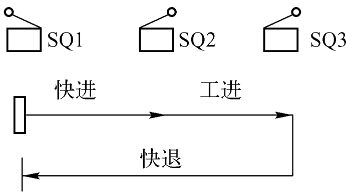


图 4-19　滑台运动示意图

3．任务实施

（1）I/O 点的分配

PLC 的 I/O 地址分配见表 4-7。

表 4-7　PLC 的 I/O 地址分配表

PLC 的 I/O 地址	连接的外部设备	在控制系统中的作用
I0.1	SB1	启动滑台工作
I0.2	SQ1	滑台在原点位置
I0.3	SQ2	滑台运动到工进起点位置
I0.4	SQ3	滑台运动到工进终点位置
Q0.0	YV1	滑台快进/滑台工进
Q0.1	YV2	滑台工进
Q0.2	YV3	滑台快退

（2）顺序功能图的绘制

如果一个控制系统可以分解为几个独立的动作或工序，且这些动作或工序按照预先设定的顺序自动执行，则称为顺序控制系统，其特点就在于一步一步按照顺序进行。这种控制系统在进行 PLC 程序设计时，可采用顺序控制设计法，根据系统工艺流程，绘制出顺序功能图，再根据顺序功能图画出梯形图。

顺序功能图也是一种通用的技术语言，很容易被初学者接受；顺序功能图主要由步、动作以及转换条件组成，也称为顺序功能图的三要素。

1）功能图的组成

① 步（状态）

顺序功能图设计法的基本思想就在于将系统的一个工作周期划分为有限个顺序相连的阶段，每个阶段均称为“步”或“状态”。每一步可用不同编号的顺序控制继电器 S（或辅助继电器 M）进行标注和区分。

步可以根据输出量的状态变化来划分，如图 4-20 所示，如可将动力滑台按照动作顺序划分为三步。步在控制系统中具有相对不变的性质，其特点在于每一步都对应于一个稳定的输出状态。

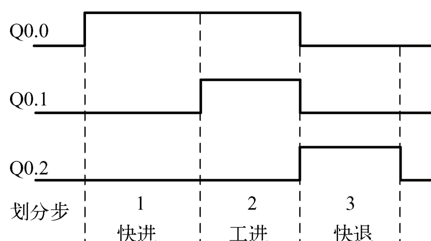


图 4-20　状态步的划分

步的图形符号如图 4-21a 所示，用矩形框表示，框中的数字是该步的编号，可采用 PLC 内部的辅助继电器 M、顺序控制继电器 S 来区分，其中初始步对应于控制系统的初始状态，是系统运行的起点。一个控制系统至少有一个初始步，初始步用双线框表示，如图 4-21b 所示。

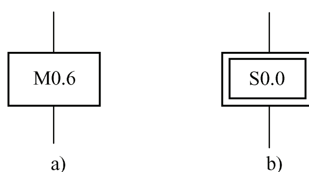


图 4-21　步的图形符号

② 动作

一个步表示控制过程中的一个稳定状态，它可以对应一个或多个动作。可以在步右边加一个矩形框，在框中用简明的文字说明该步所对应的动作，如图 4-22 所示。当该步被激活时（称其为活动步），相应的动作开始执行。

图 4-22 中 a 表示一个步对应一个动作；图 4-22b 和图 4-22c 表示一个步对应多个动作，

两种方法任选一种。

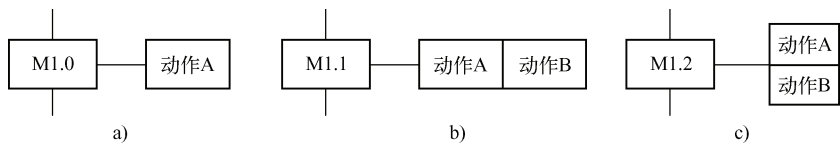


图 4-22　动作说明的表示方法

③ 转换条件

步与步之间用一个有向线段连接，表示从一个步转换到另一个步。如果表示方向的箭头是从上指向下（或从左到右），此箭头可以忽略。当系统由活动步切换到下一步时，必须满足一定的信号条件，该条件称为转换条件。转换条件将相邻的两个工作步（状态）分隔开，转换条件可以用文字、逻辑表达式或编程元件等表示。转换条件放置在短线的旁边，如图 4-23 所示。

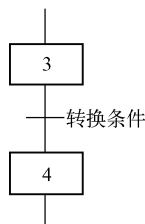


图 4-23　转换条件和有向线段的图形符号

2）功能图的绘制规则

绘制功能图时必须满足以下几点：

① 步与步之间不能直接相连，必须用一个转换条件将它们隔开。

② 转换条件与转换条件之间也不能直接相连，必须用一个步将它们隔开。

③ 顺序功能图中的初始步一般对应于系统等待启动的初始状态，这一步可能没有输出，只是做好预备状态；一个功能图至少有一个初始步。

④ 自动控制系统应能多次重复执行同一工艺过程，因此在顺序功能图中一般应有由步和有向连线组成的闭环，即在完成一次工艺过程的全部操作之后，应可以从最后一步返回初始步，再次重复执行或停止在初始状态。

⑤ 在顺序功能图中，可以用初始化脉冲 SM0.1 的动合触点作为转换条件，将初始步预置为活动步，也可以外加一个转换条件来激活初始步；否则因顺序功能图中没有活动步，系统将无法工作。

根据以上原则以及被控对象工作内容、运行步骤和控制要求，将液压滑台工作过程划分为三步，这三步状态可以用辅助继电器 M 表示，如图 4-24 所示。当某一步为活动步时，对应的辅助继电器为 1，某一转换条件实现时，该转换的后续步变为活动步（即工作步），同时该步转为不活动步。步与步之间的转换按照有向线段确定的路线进行。

绘制顺序功能图的目的是寻找某种规律或方法进行程序的编写，因此绘制功能图是顺序控制设计法最为关键的一个步骤。

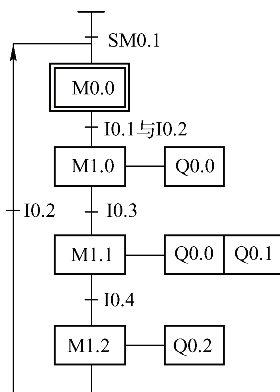


图 4-24　滑台运动顺序功能图

（3）程序的实现

根据顺序功能图，就可以按照某种编程方式编写梯形图程序，例如对图 4-24 进行梯形图编写，可以采用置位/复位指令编写程序，如图 4-25 所示，也可以采用通用逻辑指令（启保停）来编写梯形图，如图 4-26 所示。

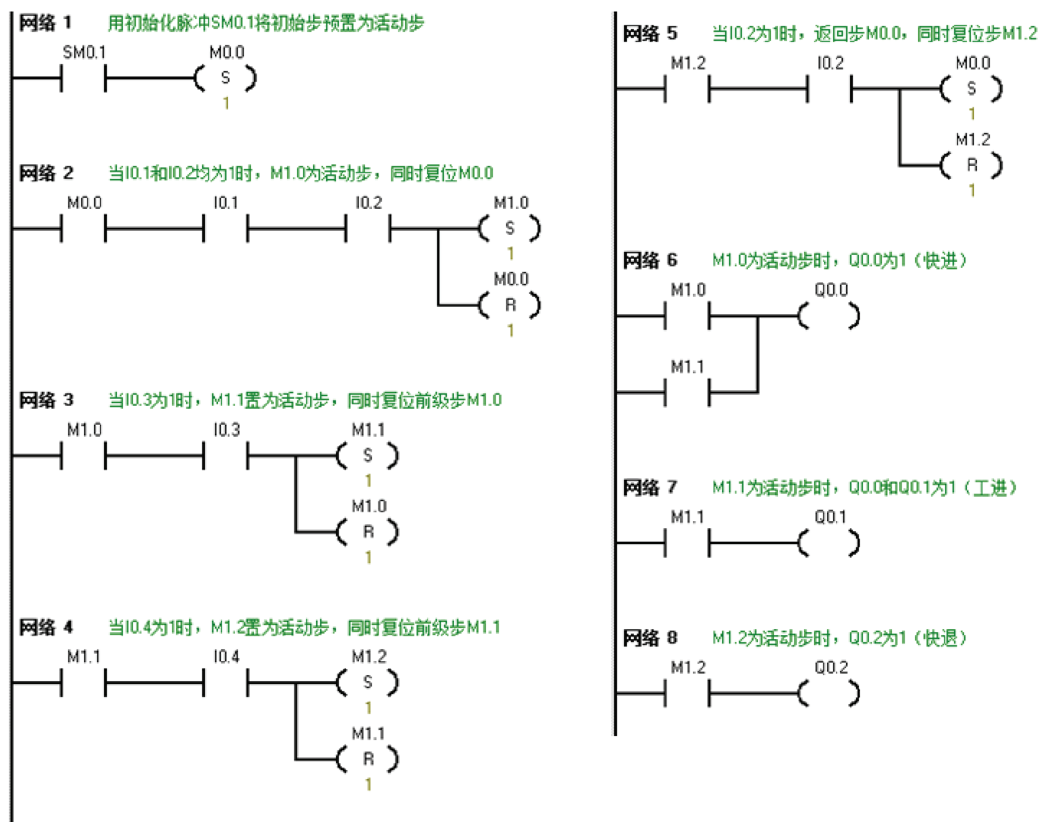


图 4-25　滑台运动梯形图（使用置位/复位指令编写程序）

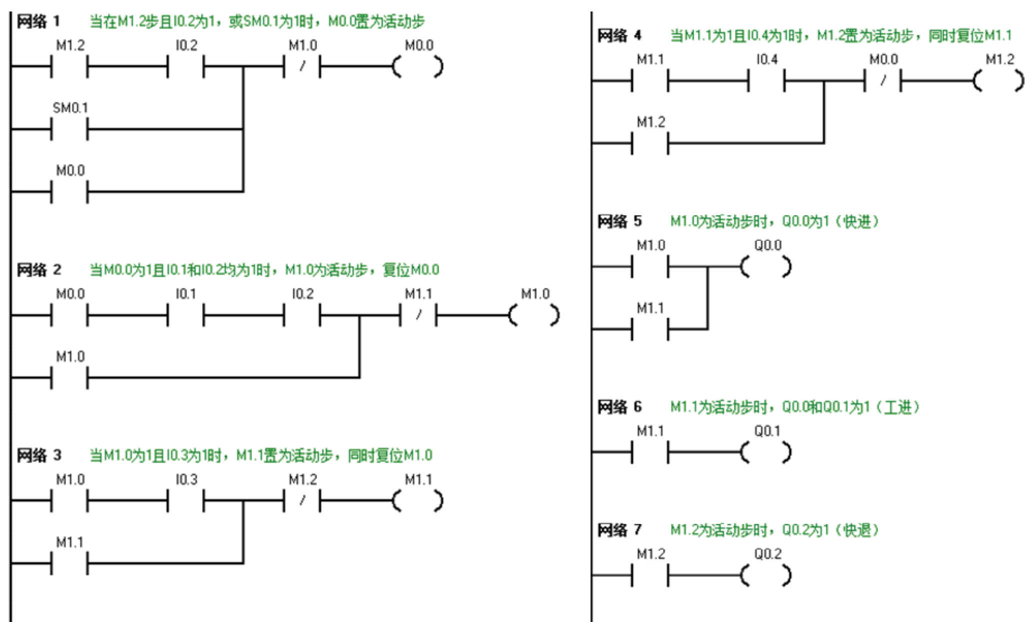


图 4-26　滑台运动梯形图（使用通用逻辑指令编写程序）

用置位/复位指令编写梯形图实际上是一种以转换条件为中心的编程方法。例如，根据功能图，M1.1 状态要想成为活动步，必须满足两个条件，一是它的前级步（即 M1.0）为活动步，二是转换条件（即 I0.3）满足。所以在梯形图中，采用 M1.0 和 I0.3 的动合触点串联电路来表示上述条件。当这两个条件同时满足时，电路接通，此时完成两个操作，该转换的后续步 M1.1 通过置位指令置位，从而变为活动步，前级步 M1.0 通过复位指令复位变为不活动步。每一步的编程都与转换实现的基本规则有着严格的对应关系，程序编写时很简单，调试时也很方便、直观。

通用逻辑指令是指 PLC 最基本的与触点和线圈有关的指令，如 LD、A、O、=等。任何一种可编程序控制器的指令系统都有这一类指令，所以这是一种通用的编程方法，可以用于任意型号的可编程序控制器。使用这种编程方法，关键是找出每一步的启动条件和停止条件，同时由于转换条件大都是短信号，需要使用具有保持功能的电路，因此这种编程方法又称为使用启保停电路的编程方法。

但对于编制复杂顺序功能图的梯形图，由于触点太多，采用这两种方法，调试和查找故障时也都显得较为麻烦和困难。

4.4.2　三台电动机顺序起停控制功能的实现

1．任务目标

设计一个顺序控制系统，要求如下：

三台电动机 M1、M2、M3，按下起动按钮时，M1 先起动；运行 2s 后 M2 起动，再运行 3s 后 M3 起动；按下停止按钮时，M3 先停止，3s 后 M2 停止，2s 后 M1 停止。

在起动过程中也应能完成逆序停止，例如在 M2 起动后和 M3 起动前按下停止按钮，M2 停止，2s 后 M1 停止。

2．任务实施

（1）电气主电路的实现

根据控制要求，完成电气主电路接线如图 4-27 所示。

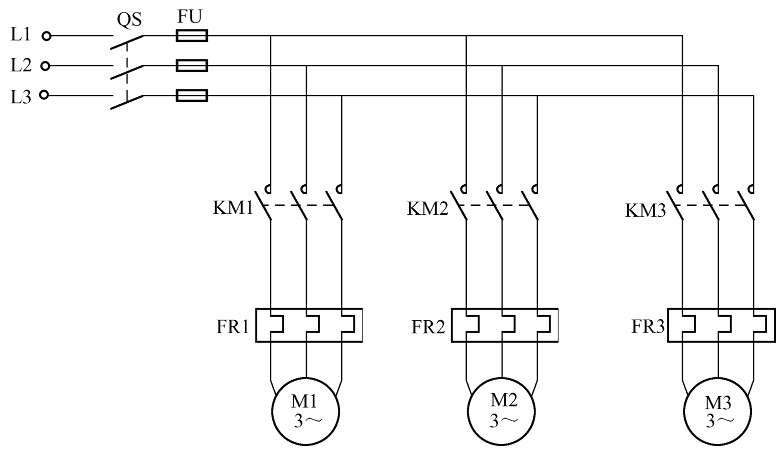


图 4-27　系统主电路

（2）PLC I/O 地址分配及接线图

PLC I/O 地址分配见表 4-8，PLC 外部接线如图 4-28 所示。需要注意两点，一是热继电器触点若按图 4-28 所示的连接形式，则热继电器 FR1～FR3 应选择手动复位方式；二是停止按钮 SB2 以常闭形式接入，编程时需要注意逻辑关系。

表 4-8　PLC 输入输出地址分配表

PLC 的 I/O 地址	连接的外部设备	在控制系统中的作用
I0.0	SB1（NO）	起动命令
I0.1	SB2（NC）	停止命令
Q0.1	KM1	第一台电动机运行
Q0.2	KM2	第二台电动机运行
Q0.3	KM3	第三台电动机运行

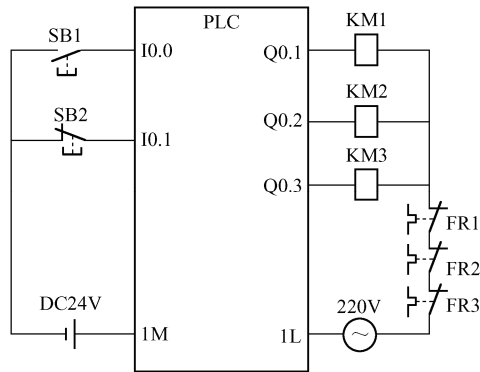


图 4-28　PLC I/O 外部接线图

（3）画出顺序功能图

绘制顺序功能图时，除了采用上面所提到的中间继电器 M，也可以用顺序控制继电器 S 来表示，采用的继电器类型不同，编写相应的梯形图的方法就不同。在使用顺控继电器时需要注意，每一个 S 位代表顺序功能图中的一步（一个状态），顺控继电器的范围为 S0.0～S31.7，共有 256 个状态。

按照控制要求，可以将三台电动机顺序起停控制系统细分为 7 步，见表 4-9。

表 4-9　三台电动机顺序起停系统步的划分

动作顺序	步号	动作	转换条件
0	S0.0	初始状态	I0.0 置 1(SB1 按下) →S1.0
1	S1.0	1#电动机起动并计时 2s	T37=1 且 I0.1=1（SB2 未按下）→S1.1 I0.1=0（SB2 按下）→S1.5
2	S1.1	2#电动机起动并计时 3s	T38=1 且 I0.1=1（SB2 未按下）→S1.2 I0.1=0（SB2 按下）→S1.4
3	S1.2	3#电动机起动	I0.1=0（SB2 按下）→S1.3
4	S1.3	3#电动机停止并计时 3s	T39=1（计时到）→S1.4
5	S1.4	2#电动机停止并计时 2s	T40=1（计时到）→S1.5
6	S1.5	1#电动机停止	Q0.1=0 →S0.0

根据以上划分，绘制顺序功能图如图 4-29 所示。这个顺序功能图既包含循环序列又包含跳步序列，它们都是选择序列的特殊形式。

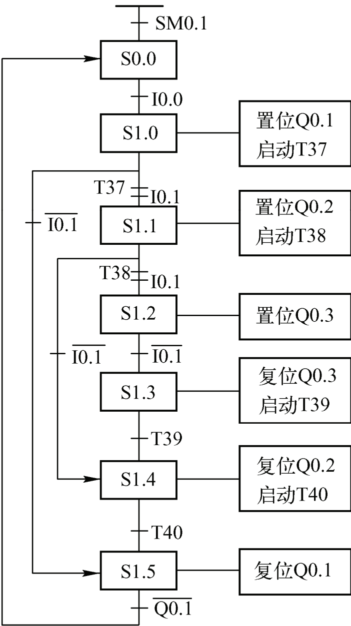


图 4-29　三台电动机顺序起停控制系统顺序功能图

（4）程序的实现

许多 PLC 都有专门用于编制顺序控制程序的步进梯形指令及编程元件。S7-200 系列 PLC 顺序控制指令及其指令功能见表 4-10。

表 4-10　顺序控制指令的形式及功能

指　令	梯 形 图	功　能	操 作 对 象
LSCR S_bit	S0.5 SCR	SCR 顺序状态开始	S（位）
SCRT S_bit	S0.6 (SCRT)	SCR 顺序状态转移	S（位）
SCRE	(SCRE)	SCR 顺序状态结束	无

从表 4-10 中可以看出，顺序控制指令的操作对象为顺控继电器 S，也把 S 称为状态继电器，每一个 S 的位都表示功能图中的一个状态。

从 LSCR 指令开始到 SCRE 指令结束之间的所有指令组成一个顺序控制继电器（SCR）段。LSCR 指令标记一个 SCR 段的开始，当该段的状态继电器置位时，允许该 SCR 段工作；SCR 段必须用 SCRE 指令结束。当 SCRT 指令的输入端有效时，一方面置位下一个 SCR 段的状态继电器 S，以便使下一个 SCR 段工作；另一方面又同时使该段的状态继电器复位，使该段停止工作。由此可以总结出每一个 SCR 程序段一般有三种功能：

1）驱动处理：即在该段状态继电器有效时，处理相应的工作；有时也可以不做任何工作。

2）指定转移条件和目标：即满足转移条件后状态转移到指定目标。

3）转移源自动复位功能：状态发生转移后，置位下一个状态的同时，自动复位原状态。

顺序功能图示例如图 4-30a 所示，梯形图如图 4-30b 所示，语句表如图 4-30c 所示。当进入 S2.0 状态时，输出 Q0.0；如果 I0.0 条件满足，置位 S2.1，进入 S2.1 状态，同时自动退出 S2.0 状态，Q0.0 复位。

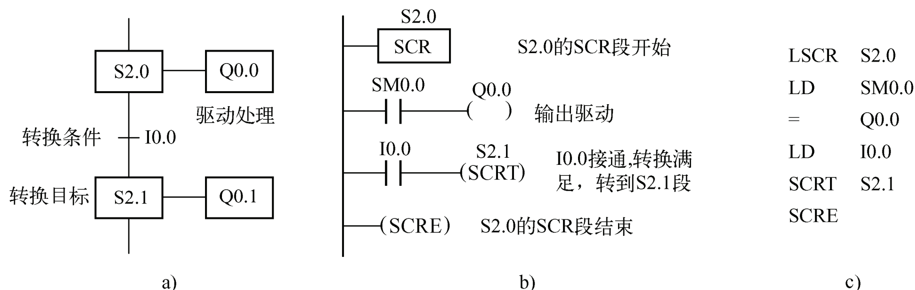


图 4-30　顺序功能图指令使用示例

根据三台电动机顺序起停控制系统的控制要求和顺序功能图，按照顺序控制继电器（SCR）指令编程方式编写梯形图程序如图 4-31 所示。

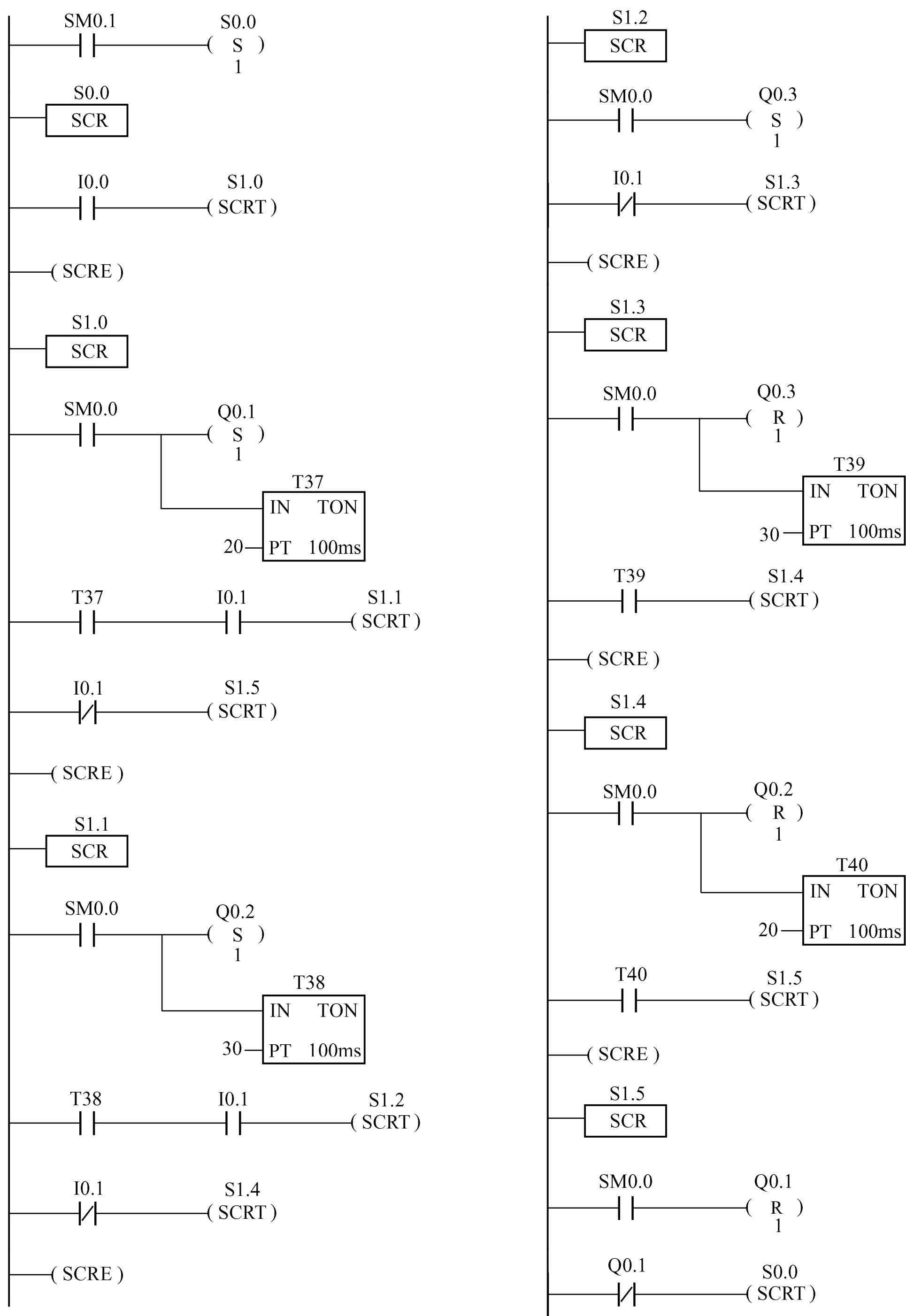


图 4-31　三台电动机顺序起停控制梯形图

4.5 综合实例

4.5.1 闪烁电路在监控系统中的应用

1．控制要求

某车间排风系统由三台风机组成，采用 S7-200 PLC 控制。其中风机工作状态需要进行监控，并通过指示灯进行显示，具体控制要求如下：

1）当系统中没有风机工作时，指示灯以 2Hz 频率闪烁。

2）当系统中只有 1 台风机工作时，指示灯以 0.5Hz 频率闪烁。

3）当系统中有 2 台以上风机工作时，指示灯常亮。

现根据以上控制要求编写风机状态监控程序。

2．PLC I/O 点的分配

通过对控制要求的分析，指示灯监控系统的输入有第一台风机运行信号、第二台风机运行信号、第三台风机运行信号 3 个输入点；输出有指示灯一个负载，占一个输出点。PLC 的 I/O 点的地址分配见表 4-11。

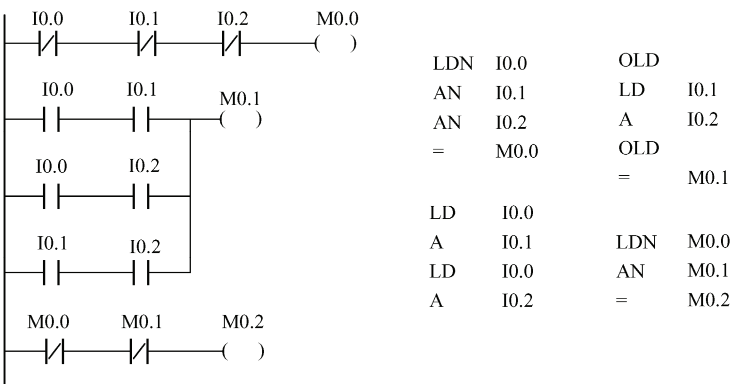
表 4-11　PLC 的 I/O 地址分配表

PLC 的 I/O 地址	连接的外部设备
I0.0	1 号风机运行信号
I0.1	2 号风机运行信号
I0.2	3 号风机运行信号
Q0.0	指示灯显示

3．程序设计

（1）风机工作状态检测程序的实现

风机工作的监视状态分为没有风机运行（M0.0）、只有一台风机运行（M0.2）和两台以上风机运行三种情况（M0.1），可以通过三个辅助继电器分别保存这三种状态，实现的程序如图 4-32 所示。




```
LDN  I0.0        OLD
AN   I0.1        LD   I0.1
AN   I0.2        A    I0.2
=    M0.0        OLD
                 =    M0.1
LD   I0.0
A    I0.1        LDN  M0.0
LD   I0.0        AN   M0.1
A    I0.2        =    M0.2
```

图 4-32　风机工作状态检测程序

（2）闪烁功能的实现

根据控制要求，需要产生 2Hz 和 0.5Hz 两种频率的闪烁信号，实现的程序如图 4-33 所示。

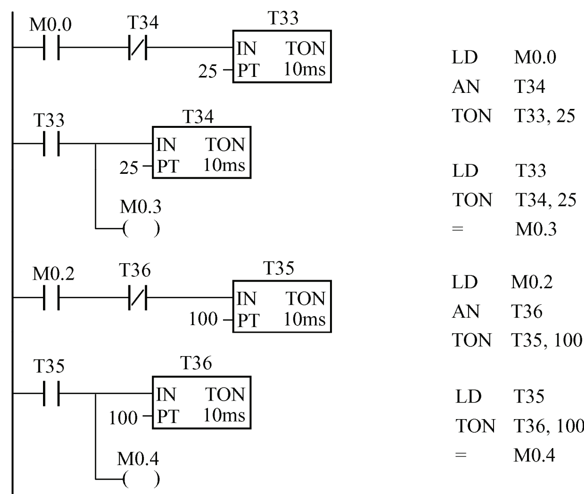


图 4-33　闪烁功能的实现

（3）指示灯输出程序的实现

指示灯输出程序需要考虑风机运行状态与对应的指示灯状态要求。当没有风机运行时（M0.0 得电），指示灯按照 2Hz 的频率闪烁（M0.3 的状态），输出指示灯启动的条件是 M0.0 的动合触点与 M0.3 的触点串联；同理，当只有一台风机运行时，输出指示灯启动的条件是 M0.2 的动合触点与 M0.4 的触点串联；由于两台以上风机运行时指示灯常亮，所以只需要用其状态显示继电器 M0.1 的动合触点驱动输出 Q0.0 就可以了，实现的程序如图 4-34 所示。

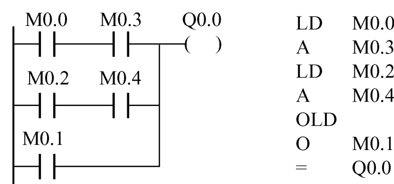


图 4-34　指示灯输出程序

为满足整个控制要求，需将以上三部分程序合并，即可构成整个监控系统的程序。

4.5.2　自动打铃程序设计与实现

1．控制要求

以 PLC 内部时钟功能编制 PLC 程序，完成某学校上午时段上/下课时间打铃功能。控制要求如下：

1）采用按钮进行北京时间校正；采用蜂鸣器打铃。

2）打铃时间：8:00、8:50、9:00、9:50、10:10、11:00、11:50，每次响铃时长为 25s。

2．预备知识

根据设计要求需要用到 PLC 内部时钟运算指令。S7-200 PLC 提供两条时钟指令分别对 CPU 的系统时钟进行读取和设置。

（1）READ_RTC（RODR）

RODR 为读取实时时钟指令，从硬件时钟读取当前时间和日期，并将其载入以地址 T 起始的 8 个字节的时间缓冲区。所有日期和格式值必须采用 BCD 格式编码，格式见表 4-12。

表 4-12　RODR 指令时间缓冲区地址及内容

地　址　号	字 节 名 称	字 节 内 容	设 定 范 围
T+0	年	当前年份	00～99
T+1	月	当前月份	01～12
T+2	日	当前日期	01～31
T+3	小时	当前小时	00～23
T+4	分钟	当前分钟	00～59
T+5	秒	当前秒	00～59
T+6	—	保留	始终设置为 00
T+7	星期	当前星期	1～7（1 是星期日，7 是星期六，0 禁用）

读取实时时钟指令的梯形图格式如图 4-35a 所示。其中 EN 为指令的使能输入端，当 I0.0 为 1 时执行该指令，T 为时钟缓冲区首地址，即 VB0 用来存放读取硬件时钟“年”的数值，依次类推，VB7 用来存放读取的“星期”数据。

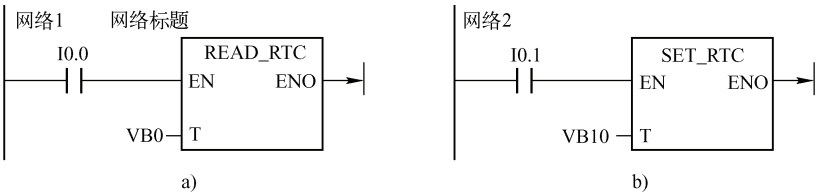


图 4-35　时钟指令盒格式

（2）SET_RTC（TODW）

TODW 为设置实时时钟指令，将当前日期和时间写入用地址 T 指定的连续 8 个字节的时间缓冲区的硬件时钟。设置实时时钟指令的梯形图格式如图 4-35b 所示。EN、T 用法同上，所不同的是，VB10 是用来设置（或改写）当前硬件时钟中“年”的数据，依次类推。

（3）使用这两条指令时的注意事项

1）由于所有读、写日期数据是 BCD 码格式，因此在线监控时按照十六进制格式查看就可得到正确的数据。

2）CPU224/CPU226 带有内置系统实时时钟，而 CPU221/222 需要外插时钟或电池卡。

3．功能实现

（1）PLC 地址分配

PLC 地址分配见表 4-13。

表 4-13　PLC 地址分配表

PLC 的 I/O 地址	连接的外部设备	在控制系统中的作用
I0.0	SB1（常开）	系统时钟校正
Q0.0	蜂鸣器	系统打铃

（续）

PLC 的 I/O 地址	连接的外部设备	在控制系统中的作用
VB0~VB7	—	设定的当前系统时钟校准值
VB10~VB17	—	当前系统时钟值

（2）程序设计

根据控制要求设计程序如图 4-36 所示。其中图 4-36a 网络 1 为系统时钟校正程序，校准值存放在 VB0 ~ VB7 共 8 个字节的存储器中，校准时间为 2014 年 03 月 07 日 08 时 0 分 0 秒，星期六，I0.0 的上升沿确保按下校准按钮后的第一个周期执行校准动作；图 4-36b 网络 2 为读取系统时间，每隔 0.5s 读取一次系统时间，并将读取结果存放在 VB10 ~ VB17 中，以便进行时间比较；图 4-36c 网络 3 为当系统时间满足打铃要求时就驱动蜂鸣器工作，并响铃 25s。

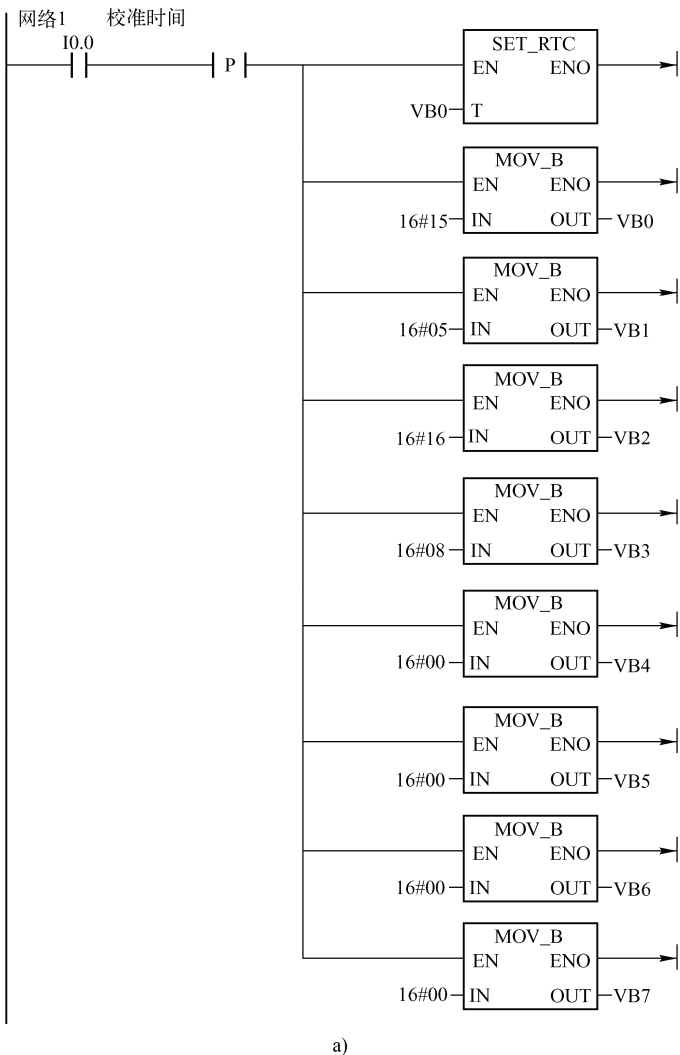
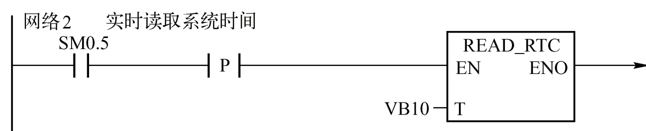
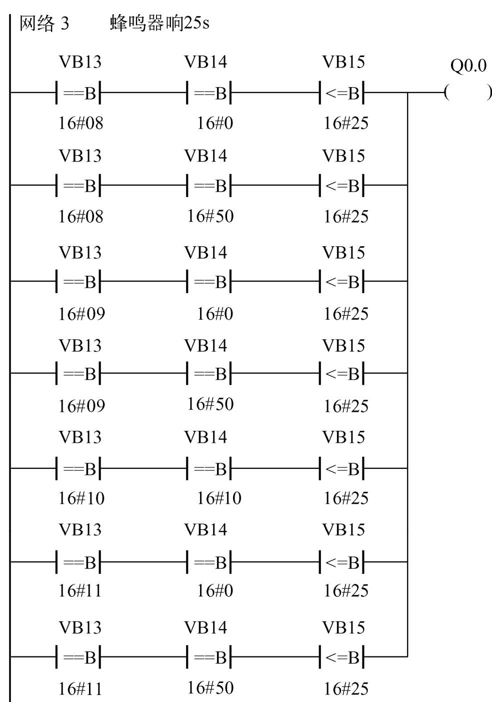


图 4-36　程序设计

a) 时钟校准



b)



c)

图 4-36　程序设计（续）

b) 读取系统时钟　c) 驱动蜂鸣器打铃

4.5.3　台车的呼车控制系统设计与实现

1．控制要求

用 PLC 实现台车的呼车控制系统的功能。如图 4-37 所示，一部电动运输车供 8 个加工点使用。台车的控制要求如下：PLC 上电后，车停在某个加工点（下称工位），若无用车呼叫（下称呼车）时，则各工位的指示灯亮，表示各工位可以呼车。如果工作人员按下本工位的呼车按钮呼车时，各工位的指示灯均灭，此时其他工位呼车无效。如停车位呼车时，台车不动，呼车工位号大于停车位号时，台车自动向高位行驶，当呼车位号小于停车位号时，台车自动向低位行驶，当台车运行到呼车工位时自动停车。停车时间为 30s 供呼车工位使用，其他工位不能呼车。从安全角度出发，停电再来电时，台车不应自行起动。

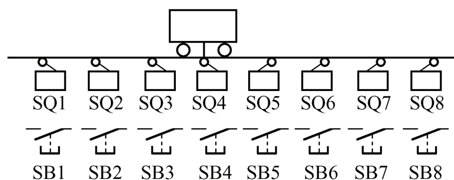


图 4-37　台车运行示意图

2．预备知识

根据设计要求，程序结构可以使用一个主程序，也可以采用在主程序中调用子程序的方法，

本设计采用调用子程序的方法，这部分内容在前面 3.4 节中已有描述，可参阅。

子程序逻辑关系编写时主要考虑比较台车实际位置与呼叫工位编号的关系。由图 4−24 可见，当台车位置小于呼叫位置时，控制台车右行；当台车位置大于呼叫位置时，控制台车左行，可用到数据比较指令，这部分内容在前面 3.5 节中已有描述，可参阅。

3．功能的实现

（1）PLC I/O 地址分配

PLC I/O 地址分配见表 4−14。

表 4−14　PLC 的 I/O 地址分配表

PLC 的 I/O 地址	连接的外部设备	PLC 的 I/O 地址	连接的外部设备
I0.0	限位开关 SQ1	I1.0	呼车按钮 SB1
I0.1	限位开关 SQ2	I1.1	呼车按钮 SB2
I0.2	限位开关 SQ3	I1.2	呼车按钮 SB3
I0.3	限位开关 SQ4	I1.3	呼车按钮 SB4
I0.4	限位开关 SQ5	I1.4	呼车按钮 SB5
I0.5	限位开关 SQ6	I1.5	呼车按钮 SB6
I0.6	限位开关 SQ7	I1.6	呼车按钮 SB7
I0.7	限位开关 SQ8	I1.7	呼车按钮 SB8
I2.0	启动按钮 SB9	I2.1	停止按钮 SB10
Q0.0	电动机制动系统	Q0.2	电动机正转，右行
Q0.1	可呼车指示灯	Q0.3	电动机反转，左行

（2）程序设计

根据控制要求设计主程序如图 4−38 所示。其中网络 1 为呼车控制系统启动、停止功能，M0.0 为系统启动/停止标志位；网络 2 为系统启动后给位置存储器 VB0、工位呼叫存储器 VB1 清零，以便下一步接收新的数据；网络 3 为系统启动后调用呼车子程序。

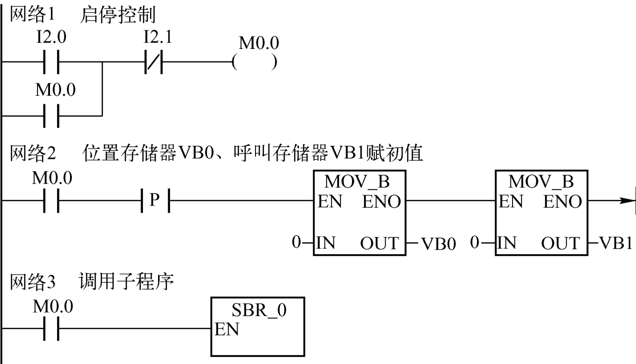


图 4−38　主程序

呼车控制子程序主要包括台车位置显示、呼叫信号处理和台车驱动处理三部分功能，程序如图 4−39 所示。网络 1～网络 8 为台车实际位置编号显示；网络 9～网络 16 为呼叫台车工位编号显示；当有呼叫信号时，呼叫按钮按下，网络 18 的 M0.1=1，从网络 17 可见，可用车指示灯 Q0.1=0，其他工位不能再呼叫台车，直到本次用车完毕，T37 动作，M0.1=0，可用车指示灯重新点亮；网络 19～网络 22 为系统判断台车位置与呼叫编号的关系，或者左行(Q0.3=1)或者右行(Q0.2=1)，直到台车到达呼叫位置，VB0=VB1，台车制动（Q0.0=1），同时延时 30s，完成工位工作；M0.2 的逻辑控制确保了当上电后台车不能自启动，符合控制要求。

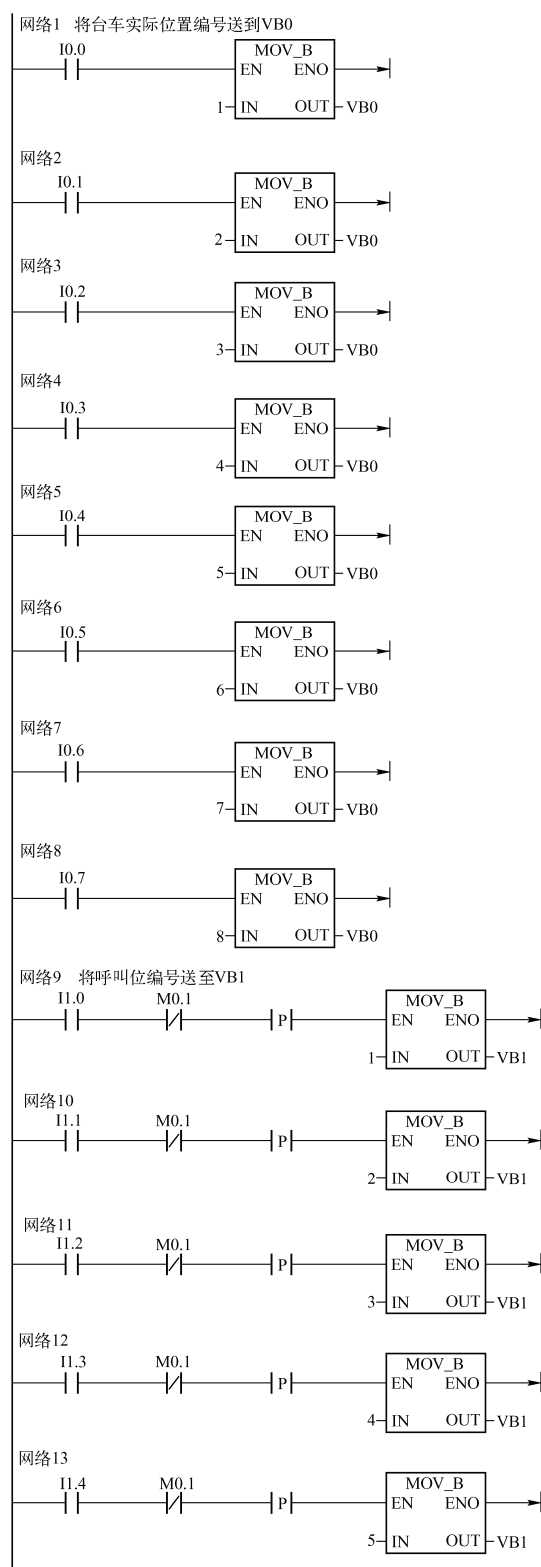


图 4-39 呼车控制子程序

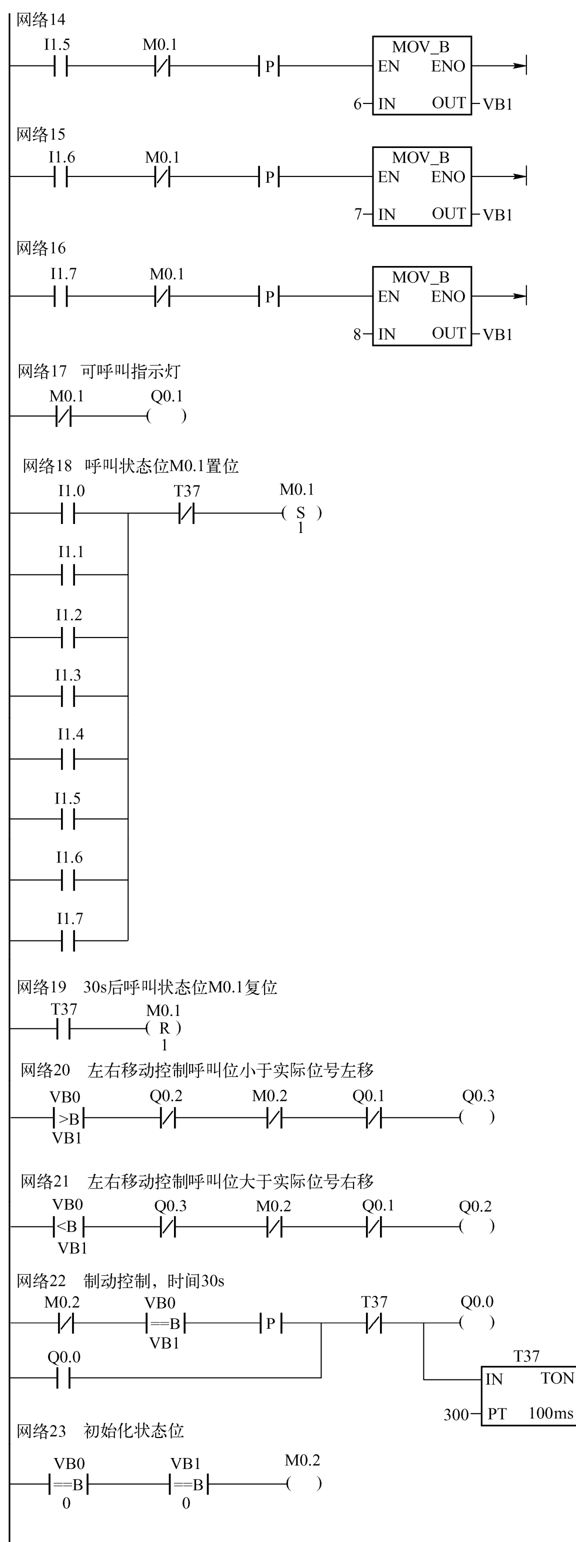


图 4-39　呼车控制子程序（续）

第 5 章　S7-200 PLC 的通信

5.1　S7-200 PLC 通信概述

S7-200 系列 PLC 内部集成的串行通信接口（编程口或 PPI 接口）为用户提供了强大的通信功能。通信接口的物理特性为 RS-485。CPU221、CPU222、CPU224 只有一个 RS-485 接口，定义为 PORT0 口；CPU224XP、CPU226 有 2 个 RS-485 接口，分别定义为 PORT0 口和 PORT1 口。使用不同的协议，通过串行通信接口可以与不同的智能设备进行通信或组成网络。

S7-200 PLC 通信接口支持以下协议：

1）PPI 协议：点对点接口协议，西门子专为 S7-200 PLC 系统开发的通信协议。

2）MPI 协议：多点接口协议（Multi Point Interface），西门子专为 S7-200/300/400 PLC 开发的通信协议，S7-200 PLC 只能作从站，因此只能被作为主站的 S7-300/400 或者其他智能设备访问。

3）自由口通信协议：本机的 PPI 接口可以由用户自定义通信协议，使 S7-200 PLC 可以与许多通信协议公开的其他设备、控制器等智能设备进行通信，如串行打印机等。

S7-200 PLC 编程软件 Micro/WIN 提供了通过自由口通信协议实现的通信功能：

1）Modbus RTU 指令库：用于 S7-200 PLC 与支持 Modbus RTU 协议的设备之间进行通信。

2）USS 指令库：用于 S7-200 PLC 与西门子传动产品（如变频器 MM440/430/420 等）之间的通信。

S7-200 支持的 PPI、MPI、自由口通信协议都是建立在 RS-485 硬件基础上的，为了保证足够的传输距离和通信速率，一般使用西门子公司制造的网络电缆和网络连接器，使用一般的双绞线连接也可以，但不一定能保证通信性能。

此外，在通信时，如果 S7-200 PLC 有两个通信口，则这两个通信口基本一样，没有区别，除非指令明确要求使用 PORT0 口或 PORT1 口；它们可以在各自不同的模式、通信速率下工作，甚至它们的口地址也可以相同。

5.2　PPI 通信协议及其应用

5.2.1　PPI 通信协议

PPI 通信协议是 S7-200 PLC 之间交换信息的专用协议，是一种点对点协议，遵循主/从方式，S7-200 系列 CPU 上集成的编程口同时就是 PPI 通信联网接口，S7-200 的一些通信模块也支持 PPI 协议，Micro/WIN 软件与 CPU 进行编程通信也通过 PPI 协议。

在使用 PPI 协议时，网络中无论连接多少台 PLC 或计算机，都只能有一个主站，其他的为从站，即只能有一个站可以主动发送数据，其他的站只能被动地接收数据或当主

站要求自己发送数据时才能够发送数据，而不能主动发送数据。主站靠 PPI 管理的共享连接来与从站通信，PPI 并不限制与任意一个从站通信的主站数量，但在一个网络中，主站不能超过 32 个。

S7-200 PLC 的 PPI 网络通信建立在RS - 485 网络的硬件基础上，因此其连接属性和需要的网络硬件设备是与其他 RS-485 网络一致的。可采用 SIEMENS 提供的 9 针 D 型插头和插座，外观如图 5-1 所示。

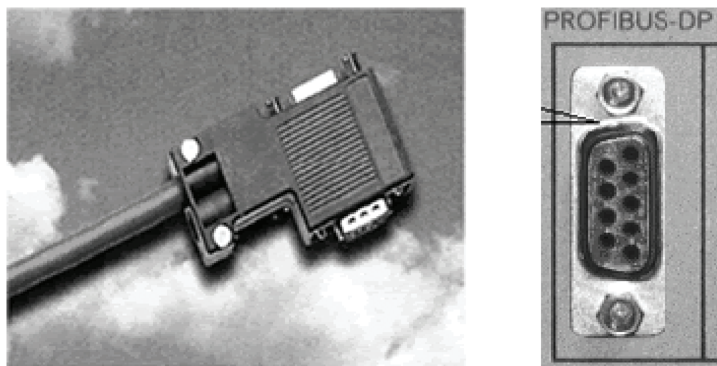


图 5-1　9 针 D 型插头和插座外观示意图

在编程时可以使用用网络读/写指令 NETR/NETW 进行数据的传递，也可以使用 Micro/WIN 中的 Instruction Wizard 中的 NETR/NETW 向导。

本节采用网络读/写指令向导进行项目设计，在 PPI 通信中只有作主站的 CPU 才需要用 NRTR/NETW 向导编程，其设置路径是：在 Micro/WIN 软件的命令菜单中选择“工具”→“向导”选项，然后在指令向导窗口中选择 NRTR/NETW 指令，如图 5-2 所示。双击图标①，弹出界面②，根据设计要求选择参数③，按下“下一步”按钮④，进入下一个界面，继续按照提示设置参数。

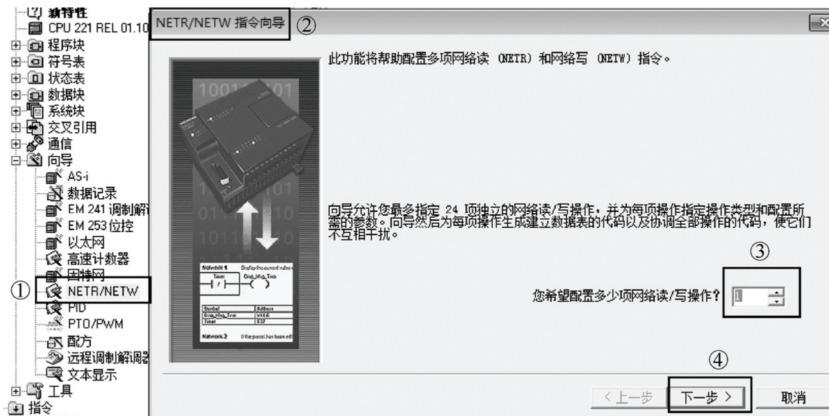


图 5-2　采用网络读/写指令向导进行 PPI 网络配置界面

按照向导设置完毕后，指令向导窗口中 NRTR/NETW 生成下拉菜单，变化如图 5-3 所示。NETR/NETW 指令向导生成的子程序管理所有的网络读/写通信，因此在编写程序时不必再编

写其他诸如设置通信口的操作等程序，调用相关指令即可。

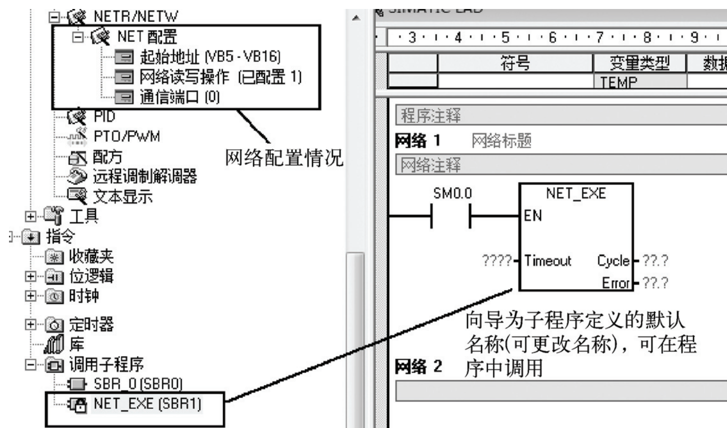


图 5-3　配置完成后菜单栏的变化

5.2.2　S7-200 PLC 之间的 PPI 通信实现

1．系统配置与操作要求

用 PPI 协议实现三台 S7-200 PLC 之间的通信，设主站为 1 号站，2 号站和 3 号站为从站，编程用的计算机的站地址为 0。要求用 1 号站的输入点 I0.0~I0.7 控制 2 号站的输出点 Q0.0~Q0.7，用 2 号站的 I0.0~I0.7 控制 1 号站的输出 Q0.0~Q0.7。用 1 号站来指定 3 号站中计数器 C4 的预定值，当计数器动作时，使 1 号主站的 Q1.1 输出为 1 状态。

三台 S7-200 系列 PLC 与装有编程软件的计算机通过 RS-485 通信接口和网络连接器，组成一个使用 PPI 协议的单主站通信网络，如图 5-4 所示，用双绞线分别将连接器的两个 A 端子连在一起，两个 B 端子连在一起。

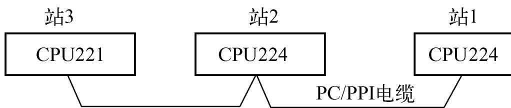


图 5-4　PPI 通信网络

2．实现 PPI 协议通信的步骤

1）网络上的每一台 PLC，设置其系统块中的通信端口参数。运行计算机上的编程软件，打开设置界面，利用 PPI/RS-485 编程电缆单独地指定 3 个站的地址（站号）和波特率。设置完成后，将系统块下载到该 PLC 中。

2）利用网络接头和网络线将各台 PLC 中用作 PPI 通信的端口 0 连接，再利用编程软件和 PPI/RS-485 编程电缆搜索出 PPI 网络中的 3 个站。

3）编写网络读/写的程序段。在 PPI 网络中，只有主站程序中使用网络读/写指令来读/写从站信息，而从站是被动的，它们不需要通信程序。在编写主站的网络读/写程序前，应首先规划好读/写数据的地址。根据要求可确定网络读/写数据规划，见表 5-1。

表 5-1　网络读/写数据规划表

1 号站（主站）	2 号站（从站）	3 号站（从站）
发送数据的长度	1 个字节	2 个字节
从主站何处发送	IB0	VB1000
发往从站何处	QB0	VB1000
接收数据的长度	1 个字节	2 个字节
数据来自从站何处	IB0	VB1010

4）利用指令向导实现网络读/写功能，过程如下：

执行菜单命令“工具”→“指令向导”，在出现对话框的第 1 页选择“NETR/NETW”（网络读/写指令），如图 5-5 所示。

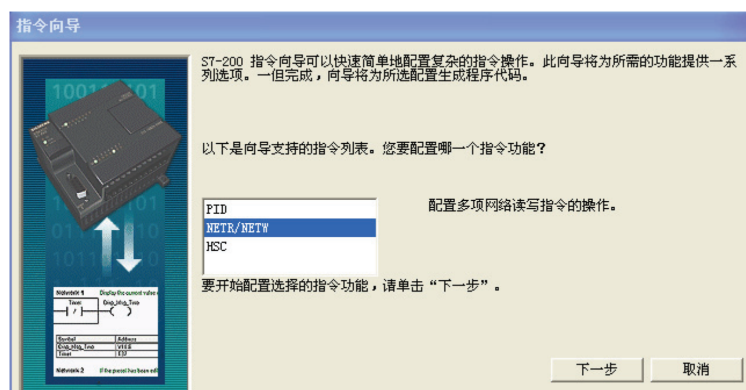


图 5-5　选择“NETR/NETW”指令

在第 2 页设置网络操作的项数为 4，操作界面如图 5-6 所示。



图 5-6　设置网络操作的项数

在第 3 页选择使用 PLC 的通信端口 0 和默认的子程序名称“NET_EXE”，如图 5-7 所示。

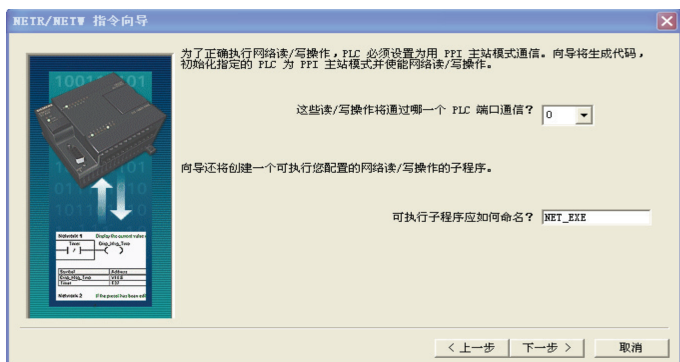


图 5-7　设置通信端口和子程序名称

在第 4 页设置操作 1 为“NETR”，要读取的字节数为 1，从地址为 2 的远程 PLC 读取它的 IB0，并存取在 1 号站 PLC 的 QB0 中，设置界面如图 5-8 所示。



图 5-8　主站从 2 号站读数据的设置

按“下一项操作>”按钮，设置操作 2 为“NETW”，将 1 号站 PLC 的 IB0 写到地址为 2 号的远程 PLC 的 QB0，设置界面如图 5-9 所示。

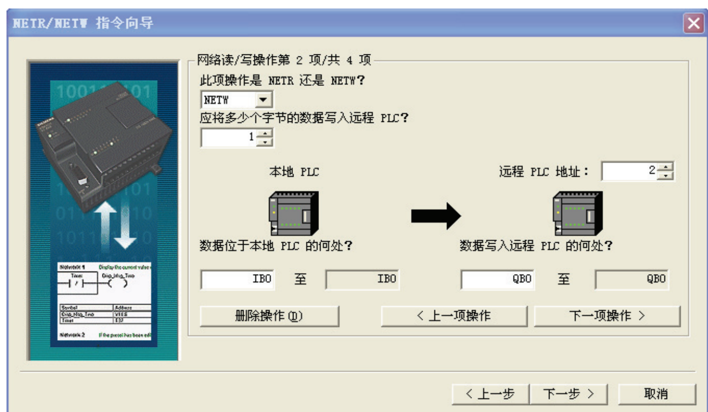


图 5-9　主站向 2 号站发送数据的设置

按“下一项操作>”按钮，设置操作 3 为“NETR”，将 3 号站 PLC 中的 VB1010~VB1011 的数据读到地址为 1 号站的远程 PLC 中的 VB1010~VB1011，设置界面如图 5-10 所示。

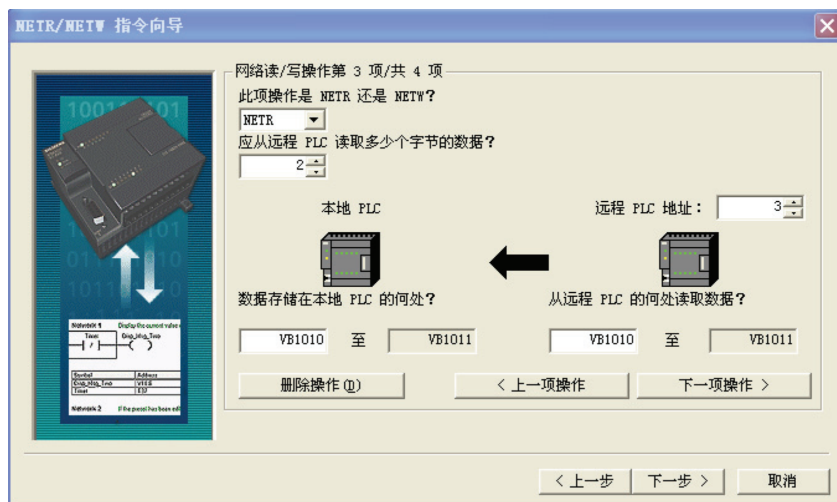


图 5-10　主站从 3 号站读数据的设置

按“下一项操作>”按钮，设置操作 4 为“NETW”，将 1 号站 PLC 中的 VB1000~VB1001 的数据写到地址为 3 号站的远程 PLC 中的 VB1000~VB1001，设置界面如图 5-11 所示。

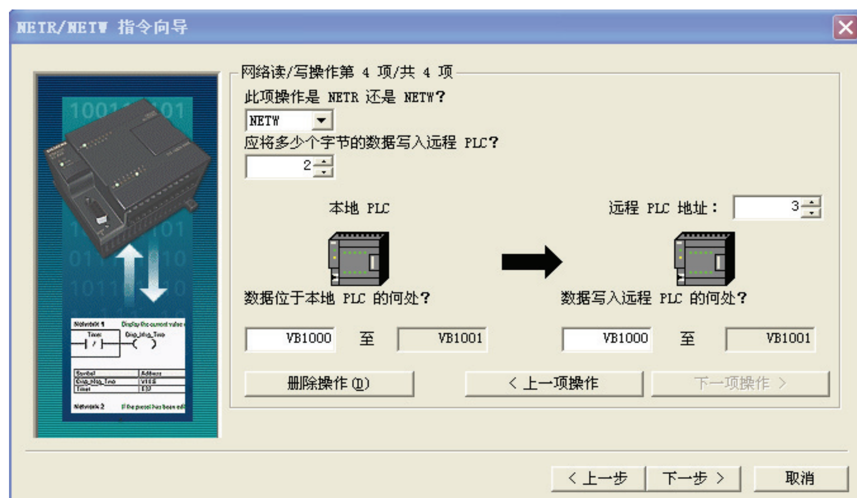


图 5-11　主站向 3 号站发送数据的设置

向导中的设置完成后，在编程软件指令树最下面的“调用子程序”文件夹中将会出现子程序 NET_EXE。在指令树的文件夹“\符号表\向导”中，自动生成了“NET_SYMS”的符号表，它给出了操作 1 和操作 2 的状态字节的地址以及超时错误标志的地址。

3．程序的实现

根据控制要求，实现程序如图 5-12 和图 5-13 所示，其中 2 号站不需要程序。NET_EXE 指令根据 NETR/ NETW 向导中设置的输入和输出执行 NETR/ NETW 功能，Timeout 表示设

定的通信超时时限，1~32767s，若为 0，则不计时；Cycle 表示输出开关量，所有网络读/写操作每完成一次都会切换“Cycle”状态；Error 表示发生错误时报警输出。

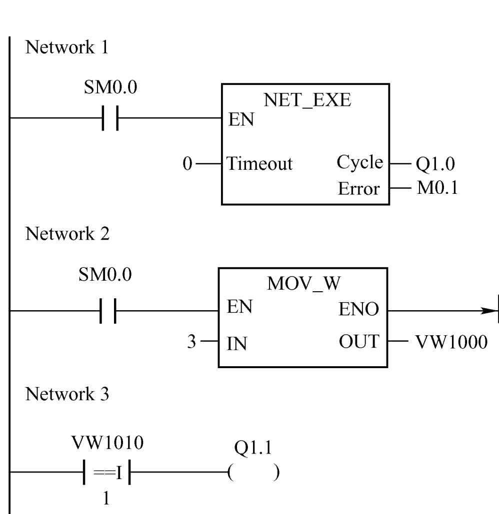


图 5-12　1 号站程序

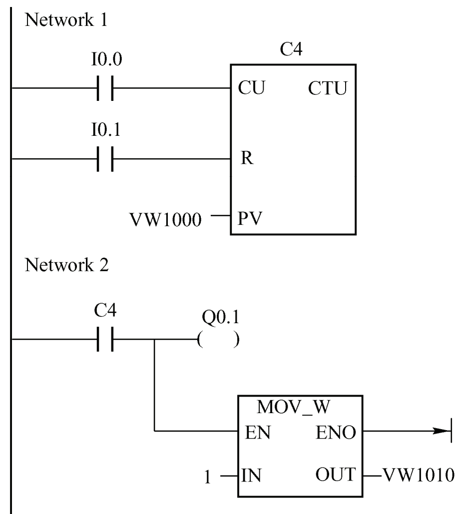


图 5-13　3 号站程序

5.3　Modbus 通信协议及其应用

5.3.1　Modbus 通信协议介绍

Modbus 是 Modicon 公司于 1979 年开发的一种通用串行通信协议，是国际上第一个真正用于工业控制的现场总线协议。由于其功能完善且使用简单、数据易于处理，因而在各种智能设备中被广泛采用。

许多工业设备包括 PLC、智能仪表等都在使用 Modbus 协议作为它们之间的通信标准。由于电器巨头施耐德公司的推动，加上相对低廉的实现成本，Modbus 现场总线在低压配电市场上所占的份额大大超过了其他现场总线，成为低压配电上应用最广泛的现场总线。Modbus 尤其适用于小型控制系统或单机控制系统，可以实现低成本、高性能的主从式计算机网络监控。2008 年 3 月 Modbus 正式成为工业通信领域现场总线技术国家标准 GB/T 19582—2008。

一个 Modbus 信息帧包括设备地址、功能代码、数据段和错误检测域。Modbus 只定义了通信消息的结构，对端口没有作具体规定，支持 RS-232、RS-422、RS-485 和以太网设备，可以作为各种智能设备、仪表之间的通信标准，方便地将不同厂商生产的控制设备连接成工业网络。

Modbus 的数据通信采用主/从方式。网络中只有一个主设备，通信采用查询-回应的方式进行，主设备初始化系统通信设置，并向从设备发送消息，从设备正确接收消息后响应主设备的查询或根据主设备的消息作出响应的动作。主设备可以是 PC、PLC 或其他工业控制设备，可以单独和从设备通信，也可以通过广播方式和所有从设备通信。单独通信时，从设备需要返回一消息作为回应，从设备回应消息也由 Modbus 信息帧构成，而以广播方式查询的，则不作任何回应。主从设备查询-回应周期如图 5-14 所示。

在图 5-14 中，查询消息中的功能代码表示被选中的从设备要执行何种功能，例如指定的从设备地址为 1，功能码为 03，则含义是要求读取 1#从站的多个寄存器值并返回它们的内容；数据段包括了从设备要执行功能的任何附加消息，例如从哪个寄存器地址开始读数据、要读的寄存器数量是多少个；错误检测域为从设备提供了一种验证消息内容是否正确的方法。

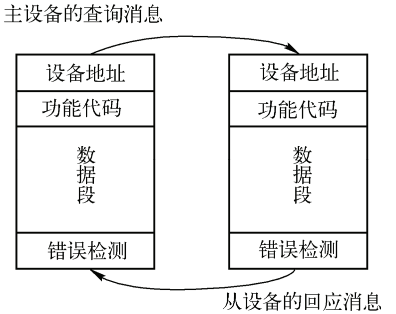


图 5-14　主从设备查询-回应周期

如果从设备产生正常的回应，则回应消息中的功能代码是对查询消息中的功能代码的回应。数据段包括了从设备收集的数据：寄存器的数据或状态，如果在消息接收过程中发生错误，或从设备不能执行其命令，从设备将建立一个错误的消息并将它作为回应发送，功能代码将被修改以指出回应消息是错误的，同时数据段包含了描述此错误信息的代码。错误检测域允许主设备确认消息内容是否可用。

Modbus 协议目前有用于串口、以太网以及其他支持互联网协议的网络的版本，大多数 Modbus 设备通信通过串口EIA-485物理层进行。由于 Modbus 协议所具有的良好的适用性，已经得到了诸如 GE、SIEMENS 等大公司的应用，并把它作为一种标准的通信接口提供给用户。

对于串行连接，在 Modbus 系统中有两种传输模式可选择，一种为 RTU（远程终端单元）模式，一种为 ASCII（美国标准信息交换代码）模式。这两种模式只是信息编码不同。RTU 模式采用二进制表示数据的方式，而 ASCII 模式使用的字符是 RTU 模式的两倍，即在相同传输速率下，RTU 模式比 ASCII 模式传输效率要提高一倍；但 RTU 模式对系统的时间要求较高，而 ASCII 模式允许两个字符发送的时间间隔达到 1s 而不产生错误。

在一个 Modbus 通信系统中只能选择一种模式，不允许两种模式混合使用，即被设置为 RTU 通信方式的节点不会和设置为 ASCII 通信方式的节点进行通信，反之亦然。通信系统选用哪种传输模式可由主设备来选择。

Modbus RTU 是一种较为理想的通信协议，也是得到最为广泛应用的工业化协议，常见的通信速率为 9600bit/s 和 19200bit/s。本节主要介绍 Modbus RTU 的基本概念及其应用。

5.3.2　Modbus RTU 通信

为了与从设备进行通信，主设备会发送一段包含设备地址、功能代码、数据段和错误检查的信息。Modbus RTU 模式下的信息传输报文格式见表 5-2。使用 RTU 模式发送消息，至少要有 3.5 个字符的时间停顿间隔作为报文的开始，这种字符时间间隔在网络波特率多样的情况下是很容易实现的。

表 5-2　RTU 信息帧报文格式

起始位	设备地址	功能代码	数据段	CRC 校验	结束
T1-T2-T3-T4	一个字节	一个字节	N 个字节	两个字节	T1-T2-T3-T4

1）设备地址：信息帧的第一个字节是设备地址码，这个字节表明由用户设置地址的从机

将接收由主机发送来的信息。每个从机都必须有唯一的地址码，并且只有符合地址码的从机才能响应回送；当从机回送信息时，相应的地址码表明该信息来自于何处。设备地址是一个 0~247 间的数字，发送给地址 0 的信息可以被所有从机接收到；但是数字 1~247 是特定设备的地址，相应地址的从设备总是会对 Modbus 信息作出反应，这样主设备就会知道这条信息已经被从设备接收到了。

2）功能码：功能码是通信传送的第二个字节，定义了从设备应该执行的命令，例如读取数据、接收数据、报告状态等（参见表 5-3），有些功能代码还拥有子功能代码。主机请求发送，通过功能码告诉从机执行什么动作；作为从机响应，从机发送的功能码与从主机得到的功能码一样，并表明从机已响应主机进行操作。功能码的范围是 1~255，有些代码适用于所有控制器，有些代码只能应用于某种控制器，还有些代码保留以备后用。

表 5-3　功能代码

功能代码	作　用	数据类型
01	读开关量输出状态	位
02	读开关量输入状态	位
03	读取保持寄存器	整型、字符型、状态字、浮点型
04	读输入寄存器	整型、状态字、浮点型
05	写单个线圈	位
06	写单个寄存器	整型、字符型、状态字、浮点型
07	读异常状态	—
08	回送诊断校验	重复回送信息
15	写多个线圈	位
16	写多个寄存器	整型、字符型、状态字、浮点型
XX	根据设备的不同，最多可以有 255 个功能代码	

3）数据段：不同的功能码，数据段的内容会有所不同。数据段包含需要从机执行什么动作或由从机采集的返送信息，这些信息可以是数值、参考地址等；对于不同的从机，地址和数据信息都不相同。例如，功能码告诉从机读取寄存器的值，则数据段必须包含要读取寄存器的起始地址及读取长度。

4）CRC 码：循环冗余校验码（CRC）是包含两个字节的错误检测码，由传输设备计算后加入到消息中，接收设备重新计算收到消息的 CRC，并与接收到的 CRC 域中的值比较，如果两值不同，表明有错误。在有些系统中，还需对数据进行奇偶校验，奇偶校验对每个字符都可用，而帧检测 CRC 应用于整个消息。

典型的 RTU 报文帧没有起始位，也没有停止位，而是以至少 3.5 个字符的时间停顿间隔标志一帧的开始或结束（见表 5-2 的 T1-T2-T3-T4）。报文帧由地址域、功能域、数据域和 CRC 校验域构成。所有字符位由十六进制 0~9、A～F 组成。

需要注意的是，在 RTU 模式中，整个消息帧必须作为一个连续的数据流进行传输。如果在消息帧完成之前有超过 1.5 个字符时间的停顿间隔发生，接收设备将刷新未完成的报文并假定下一个字节将是一个新消息的地址域；同样地，如果一个新消息在小于 3.5 个字符时间内紧跟前一个消息开始，接收设备将认为它是前一个消息的延续。如果在传输过程中有以上两种情况发生，就会导致 CRC 校验产生一个错误消息，并反馈给发送设备。

网络设备不断侦测网络总线，即使在停顿间隔时间内也不例外。当第一个域（地址域）接收到，每个设备都进行解码以判断是否是发给自己的；在最后一个传输字符之后，一个至少 3.5 个字符时间的停顿标定了消息的结束；一个新的消息可在此停顿后开始。

5.3.3　Modbus 协议的安装

在 S7-200 PLC 中，Modbus RTU 的通信协议可以通过专用指令实现，PLC 可自动生成响应帧。

Modbus 协议包含在 S7-200 PLC 的编程软件 STEP 7-Micro/ WIN 指令库（Libraries）中。当 STEP 7-Micro/WIN 安装了指令库以后，通过指令库中的 Modbus Protocol 可以打开相应的编程指令，如图 5-15 所示，Modbus 协议指令，可将 S7-200 PLC 设定为 Modbus 主站或从站进行工作。

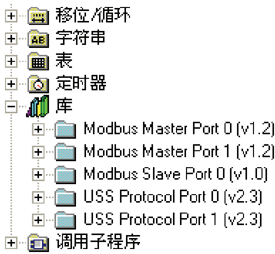


图 5-15　Modbus Protocol 指令库

指令库中有针对端口 0 和端口 1 的主站指令库 Modbus Master Port0 和 Modbus Master Port1，也有针对端口 0 的从站指令库 Modbus Slave Port0，故可利用指令库实现 S7-200 PLC 端口 0 的 Modbus RTU 主/从站通信。

西门子 Modbus RTU 协议库支持的最常用 8 条功能码含义见表 5-4。

表 5-4　西门子 Modbus RTU 协议库常用功能码

功　能　码	描　　述	说　　明
1	读取单个/多个线圈的实际输出状态	返回任意数量输出点的接通/断开状态（Q）
2	读取单个/多个线圈的实际输入状态	返回任意数量输入点的接通/断开状态（I）
3	多个保持寄存器	返回 V 存储器的内容。在一个请求中最多可读 120 个字
4	读单个/多个输入寄存器	返回模拟输入值
5	写单个线圈（实际输出）	将实际输出点设置为指定值，用户程序可以重写由 Modbus 的请求而写入的值
6	写单个保持寄存器	将单个保持寄存器的值写入 S7-200 的 V 存储器
15	写多个线圈（实际输出）	写多个实际输出值到 S7-200 的 Q 映像区。起始输出点必须是一个字节的开始（如 Q0.0 或 Q1.0），并且要写的输出数量是 8 的倍数，用户程序可以重写由 Modbus 的请求而写入的值
16	写多个保持寄存器	将多个保持寄存器写入 S7-200 的 V 区。在一个请求中最多可写 120 个字

使用 Modbus 指令库编写程序需要注意以下几点：

1）使用 Modbus 指令库前，必须将其安装到 STEP7-Micro/WIN V3.2 或以上版本的软件中。

2）S7-200 PLC 的 CPU 版本必须为 2.00 或者 2.01（即订货号为 6ES721*-*** 23-0BA*），1.22 版本之前（包括 1.22 版本）的 CPU 不支持 Modbus 指令库。

3）如果 CPU 端口被设为 Modbus 通信时，该端口就无法用于其他任何用途，包括用 STEP7-Micro/WIN 软件下载程序。

5.3.4　Modbus 地址

Modbus 地址由 5 个字符组成，包含数据类型和地址的偏移量。第 1 个字符用来指出数据

类型，后 4 个字符用来选择数据类型内的适当地址。

1. 主站寻址

Modbus 主站指令根据地址进行分类，以便完成相应的功能，并发送至从站设备。Modbus 主站指令支持下列 Modbus 地址：

00001~09999：离散输出（线圈）。

10001~19999：离散输入（触点）。

30001~39999：输入寄存器（通常是模拟量输入）。

40001~49999：保持寄存器。

所有 Modbus 地址都是从地址 1 开始编号。有效地址范围取决于从站设备的参数设置。不同的从站设备将支持不同的数据类型和地址范围。

2. 从站地址

Modbus 从站指令支持的通信内容及相应地址如下：

00001~00128：实际输出，对应于 Q0.0~Q15.7。

10001~10128：实际输入，对应于 I0.0~I15.7。

30001~30032：模拟输入寄存器，对应于 AIW0~AIW62，注意地址为偶数。

40001~4xxxx：保持寄存器，对应于 V 区。

与主站相同，所有 Modbus 地址是从地址 1 开始编号的。表 5-5 为 Modbus 地址与从站 S7-200 PLC 地址的对应关系。

表 5-5　Modbus 地址与从站 PLC 地址的对应关系

序　　号	Modbus 地址	S7-200 PLC 地址
1	00001	Q0.0
	00002	Q0.1
	…	…
	00127	Q15.6
	00128	Q15.7
2	10001	I0.0
	10002	I0.1
	…	…
	10127	I15.6
	10128	I15.7
3	30001	AIW0
	30002	AIW2
	…	…
	30031	AIW60
	30032	AIW62
4	40001	HoldStart
	40002	HoldStart+2
	…	…
	4xxxx	HoldStart+2*(xxxx-1)

Modbus 从站指令 MBUS_INIT 可以对 Modbus 主站要访问的输入、输出、模拟量输入和保持寄存器的数量、位置等进行限定。例如指令中的参数 MaxIQ 可以限定 Modbus 主站要访问的 I、Q 的最大数量；参数 MaxHold 可以限定 Modbus 主站要访问的保持寄存器的最大数量，而参数 HoldStart 则可以确定主站要访问的保持寄存器的初始位置。有关参数的用法可参考 5.3.5 节。

5.3.5 S7-200 PLC 之间的 Modbus 通信实现

1．控制要求

两台型号为 S7-200 CPU226 CN 的 PLC 进行 Modbus 通信，其中一台作为 Modbus 通信主站，另一台作为 Modbus 通信从站。当主站 I0.1 为 ON 时，主站给从站发送信息，并使从站的输出 Q0.0~Q0.7 随主站&VB1000 的值变化。

2．硬件配置

1 台 PC、2 台 PLC、1 根 PROFIBUS 网络电缆（含有两个网络总线连接器）。

3．通信的建立

如图 5-16 所示，Modbus 通信在两个 S7-200 PLC 的 Port0 通信口之间进行。选择具有两个通信口的 CPU 构成通信系统较为方便，一个作通信口用，一个与计算机连接。在主站侧可以选择 Port0 或 Port1 作 Modbus 通信口，这取决于在主站指令库中对相关指令的选择。在这里 Port1 通信口与 PC 连接，便于实现程序编制、下载和在线监控，两个 CPU 的 Port0 通信口通过 PROFIBUS 电缆进行连接，实现两台 PLC 的 Modbus 通信传输。

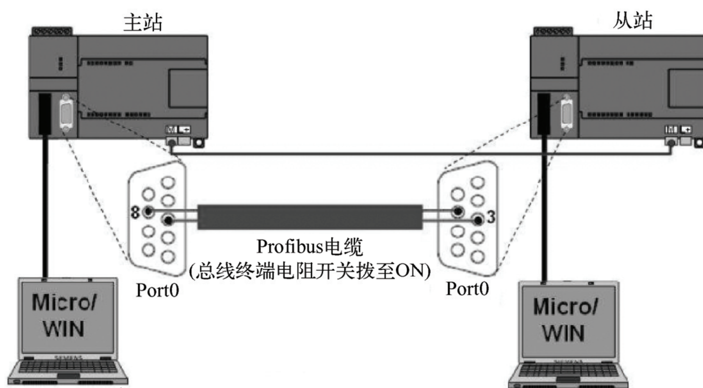


图 5-16　硬件连接

对于 Modbus 通信，主站侧需要使用 “MBUS_CTRL” 和 “MBUS_MSG”指令，从站侧需要使用“MBUS_INIT”和“MBUS_SLAVE”指令。

（1）主站侧 MBUS_CTRL 指令

MBUS_CTRL 指令如图 5-17 所示，各个参数意义见表 5-6，用于初始化主站通信，可初始化、监视或禁用 Modbus 通信。

MBUS_CTRL 指令必须在每次扫描时且“EN”输入使能时被调用，以允许监视随 MBUS_MSG 指令启动的任何突出消息的进程。指令完成后立即设定“Done”位，才能继续执行下一条指令。在使用 MBUS_MSG 指令之前，必须正确执行 MBUS_CTRL 指令。

表 5-6　MBUS_CTRL 参数选项及其意义

参　　数	意　　义	选　　项	数据类型
EN	使能		BOOL
Mode	协议选择	0—PPI，1—Modbus	BOOL
Baud	传输速率	1200，2400，4800，9600，19200，38400，57600，115200 (kbit/s)	DWORD
Parity	校验选择	0—无校验，1—奇校验，2—偶校验	BYTE
Timeout	从站的最长响应时间	1 ~32767ms；典型值是 1000ms(1s)。“超时”参数应该设置得足够大，以便从站有时间对所选的波特率作出应答	INT
Done	完成标志位	若完成输出为 1，否则为 0	BOOL
Error	错误代码	Done=1 有效时：0—无错误，1—奇偶校验选择无效，2—波特率选择无效，3—超时选择无效，4—模式选择无效	BYTE

（2）主站侧 MBUS_MSG 指令

MBUS_MSG 指令如图 5-18 所示，各个参数意义见表 5-7，用于启动对 Modbus 从站的请求并处理应答。当“EN”输入和“First”输入都为 1 时，MBUS_MSG 指令启动对 Modbus 从站的请求；通常需要多次扫描完成发送请求、等待应答和处理应答。

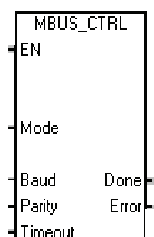


图 5-17　MBUS_CTRL 指令

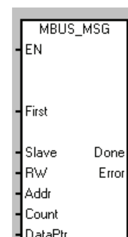


图 5-18　MBUS_MSG 指令

表 5-7　MBUS_MSG 参数选项及其意义

参　　数	意　　义	选项及说明	数据类型
EN	使能		BOOL
First	读/写请求位	在有新请求要发送时打开，以进行一次扫描	BOOL
Slave	从站地址	0~247；其中地址 0 是广播地址	BYTE
RW	读/写	0—读，1—写	BYTE
Addr	读/写从站的数据地址	00001~00128 = 数字量输出（Q0.0~Q15.7） 10001~10128 = 数字量输入（I0.0~I15.7） 30001~30032 = 模拟量输入（AIW0~AIW62） 40001~49999 = 保持寄存器	DWORD
Count	位/字的个数	地址 0xxxx：读取/写入的位数 地址 1xxxx：读取的位数 地址 3xxxx：读取的输入寄存器字数 地址 4xxxx：读取/写入的保持寄存器字数	INT
DataPtr	V 存储区起始地址指针	对于读取请求，DataPtr 指向用于存储从 Modbus 从站读取的数据的第一个 CPU 存储器位置；对于写入请求，DataPtr 指向要发送到 Modbus 从站的数据的第一个 CPU 存储器位置	DWORD
Done	完成标志位	完成输出在发送请求和接收应答时关闭。完成输出在应答完成或 MBUS_MSG 指令因错误而中止时打开	BOOL
Error	错误代码	0—无错误；1—应答时奇偶校验错误；2—未使用；3—接收超时；4—请求参数出错；5—Modbus 主设备未启用；6—Modbus 忙于处理另一个请求…	BYTE

MBUS_MSG 指令一次只能激活一条，如果启用了多条 MBUS_MSG 指令，则将处理所执行的第一条 MBUS_MSG 指令，其后的所有 MBUS_MSG 指令将被中止并产生错误代码 6。

（3）从站侧 MBUS_INIT 指令

MBUS_INIT 指令如图 5-19 所示，各个参数意义见表 5-8，用于启用和初始化或禁止 Modbus 通信。指令完成后立即设定“Done”位，才能继续执行下一条指令。应当在每次通信状态改变时执行 MBUS_INIT 指令，因此“EN”输入采用一个上升沿或下降沿打开，或者仅在首次扫描时执行。在使用 MBUS_SLAVE 指令之前，必须正确执行 MBUS_INIT 指令。

表 5-8　MBUS_INIT 参数选项及其意义

参数	意义	选项及说明	数据类型
EN	使能		BOOL
Mode	接口通信模式选择	0—PPI，1—Modbus	BOOL
Baud	传输速率	1200，2400，4800，9600，19200，38400，57600，115200（kbit/s）	DWORD
Addr	从站地址	1~247	BYTE
Parity	奇偶校验设定	0—无校验，1—奇校验，2—偶校验	BYTE
Delay	报文延迟时间	0~32760ms；默认值为 0	WORD
MaxIQ	可使用的最大数字输入输出点数	0~128，　建议使用的 MaxIQ 数值是 128，该数值可在 S7-200 中存取所有的 I 和 Q 点	WORD
MaxAI	可使用的最大模拟量输入字数	最大 AI 字数，参与通信的最大 AI 通道数，可为 16 或 32	WORD
MaxHold	最大保持型变量寄存器字数	例如，为了允许主设备存取 2000 个字节的 V 存储器，将 MaxHold 设为 1000 个字的数值（保持寄存器）	WORD
HoldStart	保持型变量寄存器的起始地址	该数值一般被设为 VB0，因此 HoldStart 参数被设为 &VB0（VB0 地址）	WORD
Done	初始化完成标志	初始化成功后置 1	BOOL
Error	出错代码	0—无错误；1—内存范围错误； … 10—从属功能未启用	BYTE

（4）从站侧 MBUS_SLAVE 指令

MBUS_SLAVE 指令如图 5-20 所示，各个参数意义见表 5-9，用于为 Modbus 主设备发出请求服务。在每次扫描且“EN”输入使能时执行该指令，以便检查和回答 Modbus 请求。MBUS_SLAVE 指令无输入参数，当 MBUS_SLAVE 指令对 Modbus 请求作出应答时，“Done”置为 1，如果没有需要服务的请求时，“Done”置为 0。“Error”输出包含执行指令的结果，该输出只有在“Done”为 1 时才有效，否则错误参数不会改变。

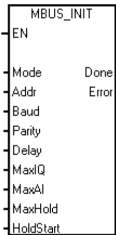


图 5-19　MBUS_INIT 指令

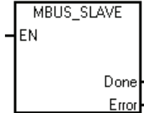


图 5-20　MBUS_SLAVE 指令

表 5-9　MBUS_SLAVE 参数意义

参　　数	意　　义	备　　注	数 据 格 式
EN	使能		BOOL
Done	完成标志位	Modbus 执行通信中时置 1，无 Modbus 通信活动时为 0	BOOL
Error	错误代码	0—无错误；1—内存范围错误；…；10—从属功能未启用	BYTE

4．功能的实现

1）编写作为 Modbus 主站的 S7-200 CPU 的 PLC 程序，将程序下载到主站 PLC 中。

如图 5-21 所示，网络 1 用于每次扫描时调用 MBUS_CTRL 指令，初始化和监视 Modbus 主站设备。Modbus 主设备设置为 9600bit/s、奇校验、允许从站延时 1ms 应答时间。

如图 5-22 所示，网络 2 实现在 I0.1 正跳变时执行 MBUS_MSG 指令，将地址 VB1000 的值写入从站 5 的保持寄存器中。

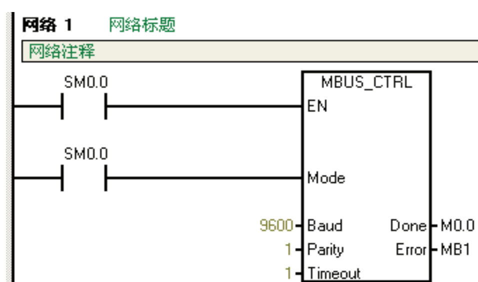


图 5-21　主站程序-网络 1

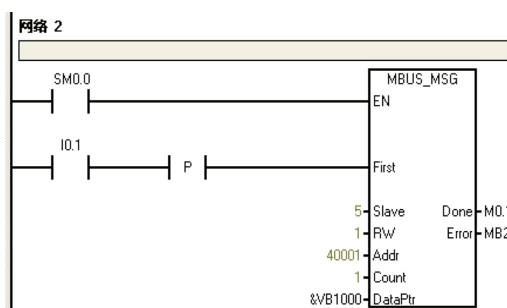


图 5-22　主站程序-网络 2

参数“DataPtr”代表了 V 区被读的起始地址，设为 VB1000，即主站读取 VB1000 的值并写入地址为 “40001”的保持寄存器中。保持寄存器以字为单位，与从站的 V 区地址对应。

如图 5-23 所示，网络 3 为给 VB1000 存储器赋初值，使其低 4 位为 1，以便监视从站的变化。

需要注意的是，利用指令库编程前首先应为其分配存储区，否则 STEP 7-Micro/WIN 编译时会报错。具体方法如下：

① 执行 STEP 7-Micro/WIN 菜单命令“文件”→“库存储区”，打开“库存储区分配”对话框，如图 5-24 所示。

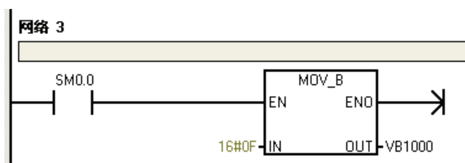


图 5-23　主站程序-网络 3

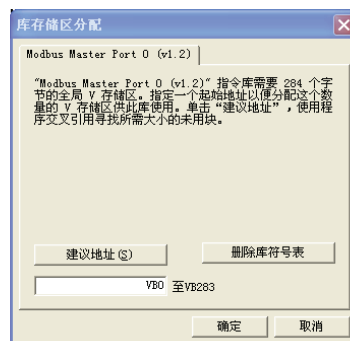


图 5-24　主站库存储区分配

② 在“库存储区分配”对话框中输入库存储区的起始地址 VB0，注意避免该地址和程序中已经采用或准备采用的其他地址重合。

③ 单击“建议地址”按钮，系统将自动计算存储区的截止地址。

④ 单击“确定”按钮确认分配，关闭对话框。

2）编写作为 Modbus 从站的 S7-200 CPU 的 PLC 程序，将程序下载到从站 PLC 中。

如图 5-25 所示，网络 1 用于初始化 Modbus 从站，即将从站地址设为 5，将端口 0 的波特率设为 9600bit/s、奇校验、延迟时间为 0； MaxIQ 取值 128、MaxAI 取值 32，表明允许存取所有的 I、Q 和 AI 数值；可以使用的 V 区寄存器地址字数设为 1000，起始地址为 VB1000，即主站的保持寄存器 40001 的值写入从站的&VB1000 中。

如图 5-26 所示，网络 2 用于在每次扫描时执行 Modbus_Slave 指令，以便响应主站报文。

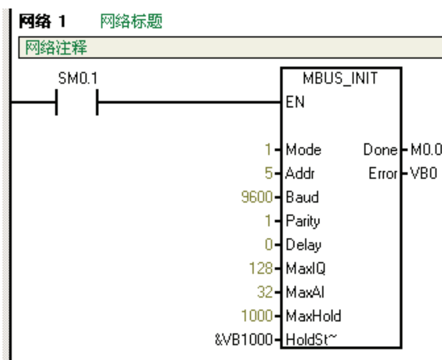


图 5-25 从站程序-网络 1

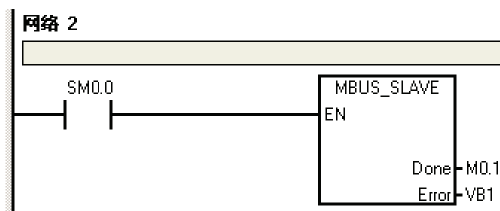


图 5-26 从站程序-网络 2

如图 5-27 所示，网络 3 为将主站传给从站的值传给 QB0，使得输出 Q0.0~Q0.7 受主站的控制，满足控制要求。

同样需要注意的是，利用指令库编程前首先应为其分配存储区，否则 STEP 7-Micro/WIN 编译时会报错，如图 5-28 所示。

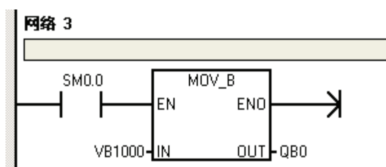


图 5-27 从站程序-网络 3

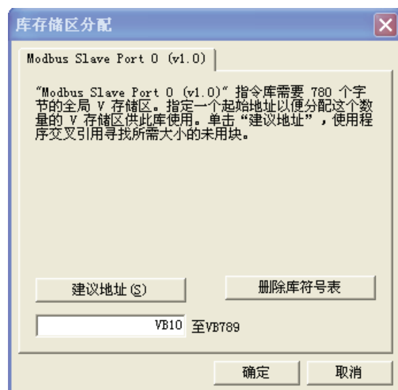


图 5-28 从站库存储区分配

3）用串口电缆连接 Modbus 主从站，观察 Modbus 从站 PLC 的 Q0.0~Q0.7 输出指示灯，或者在 Modbus 从站的 STEP 7-Micro/WIN 状态表中观察 Q0.0~Q0.7 的数值。操作步骤如下：

① 用串口电缆连接主从站 PLC 的 Port0。
② 将主从站 PLC 设置为 RUN 状态。
③ 将主站 I0.1 的开关闭合，使其状态为 ON。
④ 利用 STEP 7-Micro/WIN 状态表，在线监控从站 QB0 的数值，如图 5-29 所示。

	地址	格式	当前值	新值
1	VB1000	十六进制	16#0F	
2	QB0	十六进制	16#0F	
3		有符号		
4		有符号		
5		有符号		

图 5-29　从站状态监控表

从图 5-29 中可以看出，当主站的 I0.1 使能后，主站 VB1000 中的数据就被发送到从站，并写入从站的 VB1000 中。

5．系统设计要点

1）必须保证主站与从站的“Baud”和“Parity”的参数设置一致，并且 “MBUS_MSG”指令中的“Slave”参数要与“MBUS_INIT”中的“Addr”参数设置一致。

2）注意在 STEP7-Micro/WIN 中定义库的存储地址。

3）在从站的“MBUS_INIT”指令中，参数“HoldStart”确定了与保持寄存器起始地址 40001 相对应的 V 存储区初始地址。从站的 V 区目标指针可以这样计算：

2*(Addr-40001)+HoldStart=2*(40001-40001)+&VB1000=&VB1000

4）在从站的“MBUS_INIT”指令中，参数“MaxHold”设置的数据区要能够包含主站侧所要写入的全部数据。

5.4　USS 通信协议及其应用

5.4.1　USS 通信协议及其相关指令

1．USS 协议

USS（Universal Serial Interface Protocol，通用串行接口协议）是西门子公司为其变频器开发的通信协议，用于变频器与 PC 或 PLC 之间的通信连接，适合于规模较小的自动化系统。

USS 按照串行总线的主/从通信原理来确定访问的方法，总线上可以连接一个主站和最多 31 个从站，主站根据通信报文中的地址字符选择要传输数据的从站，在主站没有要求从站进行通信时，从站本身是不能发送数据的，各个从站之间也不能直接进行信息的传输。

PLC 作主站时，可以连接最多 31 个变频器作为通信链路中的从站。S7-200 PLC 可将其通信端口设置为自由口模式的 USS 通信协议，以便实现 PLC 对变频器的控制。

2．自由口通信模式

S7-200 PLC 在自由口通信模式下主要使用发送指令 XMT 和接收指令 RCV。通过两条指令以及相应的特殊寄存器，在自由口通信模式下可以实现数据的发送和接收。

XMT 指令用于将数据缓冲区的数据从指定的通信端口发送出去，该指令一次可发送 1~255 字节数据，指令格式如图 5-30a 所示；当允许输入 EN 有效时，初始化通信操作，通过

端口 PORT0 或 PORT1 将首地址为 TBL 的数据（发送数据缓冲区）发送到远程设备。

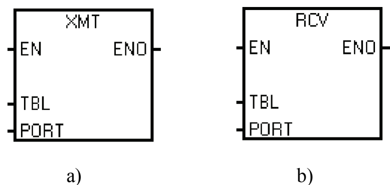


图 5-30　自由口通信指令

RCV 指令用于从指定的通信端口启动或者终止接收消息，并将接收到的消息存储在数据缓冲区中，该指令一次可接收 1~255 个数据，指令格式如图 5-30b 所示；当允许输入 EN 有效时，初始化通信操作，通过端口 PORT0 或 PORT1 接收远程设备数据，并将数据存放在首地址为 TBL 的数据缓冲区。

可以采用 XMT、RCV 指令编写 USS 通信程序，实现对变频器的监视和控制。但为了简化 S7-200 PLC 一些高级功能的使用，西门子为用户编写了多种应用子程序，并以指令库的形式提供给用户。如 USS 协议指令库、Modbus 协议指令库等；其中 USS 协议指令库是使用 USS 协议进行通信编程的方便指令，使用该指令用户不需要详细了解 USS 协议格式，仅通过简单的调用就可实现 USS 协议通信。

3．S7-200 PLC 的 USS 指令库

USS 协议包含在 S7-200 PLC 的编程软件 STEP 7-Micro/ WIN 指令库（Libraries）中。当 STEP 7- Micro/WIN 安装了指令库以后，通过指令库中的 USS Protocol 可以打开相应的编程指令，如图 5-31 所示。USS 协议指令可将 S7-200 PLC 端口 0 或端口 1 设定为 USS 工作模式，并激活相应地址的变频器从站工作；使用指令库编程用户不需要详细了解 USS 协议格式，通过简单的调用即可实现 USS 协议通信；PLC 将用这些指令库指令来控制变频器的运行和参数的读写操作。

指令库中有针对 PLC 端口 0 和端口 1 的指令库 USS Protocol Port0 和 USS Protocol Port1，故可利用指令库实现 S7-200 PLC 端口 0 或端口 1 与变频器的 USS 主/从站通信功能。

S7-200 PLC 与变频器之间的 USS 通信，可以用到的指令有 8 条，至少需要用到以下两条指令：

（1）初始化指令 USS_INIT

初始化指令 USS_INIT 用于使能或禁止 PLC 和变频器之间的通信，在执行其他 USS 协议之前必须先成功执行一次 USS_INIT 指令，只有当该指令成功执行且其完成位（DONE）为 1 后，才能继续执行下面的指令。符号如图 5-32 所示。

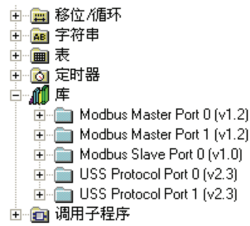


图 5-31　USS Protocol 指令库

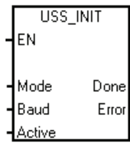


图 5-32　USS_INIT 指令

指令中各个参数意义见表 5-10。

表 5-10　USS_INIT 参数意义及其说明

参数	意义	说明	数据类型
EN	使能	通常采用脉冲指令，该位为 1 时 USS_INIT 指令被执行	BOOL
Mode	协议选择	0—PPI，1—USS	BYTE
Baud	传输速率	1200，2400，4800，9600，19200，38400，…115200 (kbit/s)	DWORD
Active	激活的从站地址	编程时只激活一次就将所有地址激活。32 位分别对应通信链路上的变频器地址 0~31。例如，若 Active 的给定值为 16#0000 0000 0000 0100，则表示链路上的通信地址为 2 的变频器被激活	DWORD
Done	完成标志位	若完成输出为 1，否则为 0	BOOL
Error	错误代码	用于存放错误代码：0—没有出错；1—变频器不能响应；3—检测到变频器响应中包含加和校验错误；…，20—在变频器中响应中的参数或数值有误	BYTE

（2）控制指令 USS_CTRL

控制指令 USS_CTRL 用于控制已经用 USS_INIT 激活的变频器，每台变频器只能使用 1 条这样的指令。该指令将用户命令存放在缓冲区内，如果指令参数 Drive 指定的变频器已经被激活，缓冲区内的命令将被发送到指定的变频器。符号如图 5-33 所示。

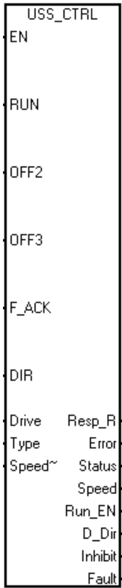


图 5-33　USS_CTRL 指令

指令中各个参数的意义见表 5-11。

表 5-11　USS_CTRL 参数意义及其说明

参　数	意　义	说　明	变量类型	数据类型
EN	使能	该位为 1 时，USS_CTRL 指令被执行，通常总是处在接通状态	IN	BOOL
RUN	变频器状态	该位为 1，且 OFF2=0、OFF3=0 时变频器以规定的速度和方向运行；该位为 0，变频器停止	IN	BOOL

（续）

参　　数	意　　义	说　　明	变 量 类 型	数 据 类 型
OFF2	变频器停止	该位为 1，则变频器自由停车	IN	BOOL
OFF3	变频器停止	该位为 1，则变频器快速制动停车	IN	BOOL
F_ACK	故障确认	用来确认故障。当该位从低到高时，变频器清除故障	IN	BOOL
DIR	变频器运行方向	0—逆时针方向；1—顺时针方向	IN	BOOL
Drive	变频器地址	用于设定变频器的站地址，有效地址 0~31	IN	BYTE
Type	变频器类型	1—MM4/G110；0—MM3 或更早产品	IN	BYTE
Speed_SP	速度设定值	该速度为全速度百分值的变频速度（−200% ~200%）	IN	REAL
Resp_R	从站应答确认信号	该位确认来自变频器的响应。对所有激活的变频器轮询最新的变频状态信息；每当 CPU 从变频器收到一个响应，该位就接通一个扫描周期，更新 ERR 中的信息代码	OUT	BOOL
Error	错误代码	用于存放错误代码，包含与变频器通信请求的最新结果	OUT	BYTE
Status	变频器返回值	存放来自变频器的状态字，反映当时的实际运行状态	OUT	WORD
Speed	变频器实际运行速度	该速度为全速度百分值的变频速度（−200%~200%）	OUT	REAL
Run_EN	变频器实际运行状态	1—变频器正在运行；0—变频器停止	OUT	BOOL
D_Dir	变频器返回的运行方向	0—逆时针方向；1—顺时针方向	OUT	BOOL
Inhibit	变频器返回的禁止状态信号	0—不禁止；1—被禁止。要消除禁止位，Fault 位必须断开，RUN、OFF2、OFF3 输入都必须断开	OUT	BOOL
Fault	变频器的故障位状态	0—无故障；1—故障。若变频器故障，变频器上会显示故障代码	OUT	BOOL

5.4.2　基于 HMI 的 S7-200 PLC USS 通信功能实现

1．控制要求

系统通过 HMI 来控制变频器的启动和停止状态，并通过 HMI 来改变变频器的运行频率。当按下 HMI 的启动按钮，变频器应根据所设定的频率运行；当改变 HMI 上的频率设定值，变频器的运行频率应随之改变；当按下 HMI 上的停止按钮，变频器停止运行。

2．控制系统硬件配置

根据控制要求，可选择 CPU226 PLC、MM440 变频器及 TP177A HMI，设备之间采用 RS-485 屏蔽双绞线通信电缆连接，硬件连接如图 5−34 所示。

（1）MM440 变频器介绍

MM440 系列是德国西门子公司广泛用于工业场合的多功能标准变频器，是用于控制三相交流电机的速度和转矩的变频器。此系列有多种型号，额定功率为 120W~250kW。

变频器的站地址必须正确设置，并与 USS 主站配置的从站地址相一致。该种变频器常用操作模式有以下几种：

1）BOP 面板操作。一般先设定 P0010=30，P970=1，将其他参数复位，然后设定 P0010=0，P0700=1，P1000=1。

2）外部（通过端子排）输入控制。一般先设定 P0010=30，P970=1，将其他参数复位，然后设定 P0010=0，P0700=2，P1000=2。

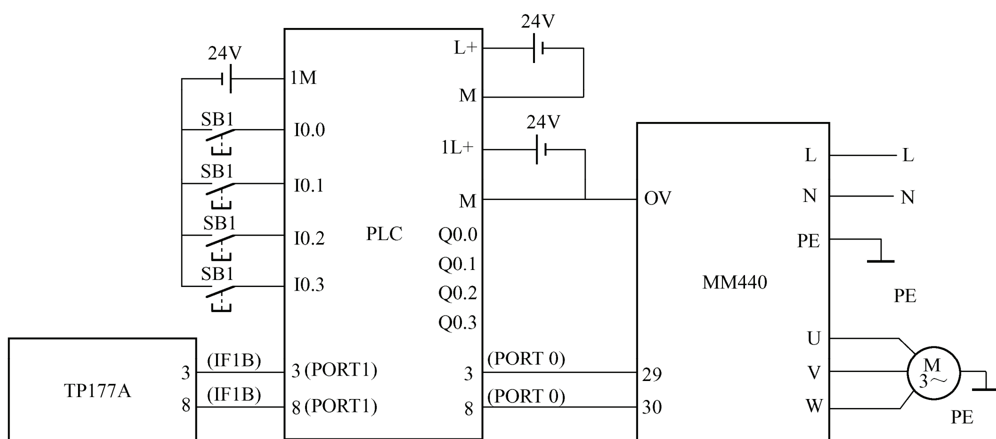


图 5-34　电气原理图

3）USS 总线控制。一般先设定 P0010=30，P970=1，将其他参数复位，然后设定 P0010=0，P0700=5，P1000=5。

4）PROFIBUS 总线控制。一般先设定 P0010=30，P970=1，将其他参数复位，然后设定 P0010=0，P0700=6，P1000=6。

其中，参数 P0010=0 变频器才能运行，若为 1 可以修改电动机主要参数；P700 为命令给定源选择；P1000 为频率给定源选择。

本例中将变频器设置为 USS 总线控制状态，且波特率为 9600bit/s（P2010.0=6）、站地址为 0（P2011.0=0）

（2）HMI 介绍

西门子 HMI 使用通用的软件工具 SIMATIC WinCC flexible 组态，适用于不同性能的面板，常见面板性能见表 5-12。

表 5-12　常见面板性能

型号 / 性能	通用面板			多功能面板		移动面板	
	77 系列	177 系列	277 系列	MP277	MP377	Mobile177	Mobile277
操作							
触摸屏		√ 1)	√ 1)	√ 1)	√ 1)		
按键	√	√ 1)	√ 1)	√ 1)	√ 1)		
触摸屏和按键		√ 2)				√	√
接口							
PPI	√	√	√	√	√	√	√
PROFIBUS	√	√	√	√	√	√ 1)	√
PROFINET		√ 2)	√	√	√	√ 1)	√ 3)
USB	√ 2)	√ 2)	√	√	√		√

（续）

性能 \ 型号	通用面板			多功能面板		移动面板	
	77 系列	177 系列	277 系列	MP277	MP377	Mobile177	Mobile277
HMI 功能							
报警系统	√	√	√	√	√	√	√
配方	√2)	√2)	√	√	√	√	√
归档			√	√	√		√
VB 脚本			√	√	√		√
软件选项		√2)	√	√	√	√	√
备注 1）不同设备类型；2）高性能产品具有的特性；3）移动面板 277 IWLAN 支持无线以太网							

SIMATIC 面板操作模式可以用键盘也可以用触摸屏，通信接口默认情况下是 PROFIBUS，还可以通过附加的接口如 USB 使得机器能与打印机等更多设备连接。

WinCC flexible 是 Protool 和 Protool/Pro 的升级产品，现有的 Protool 项目与之兼容，而且可以很方便地移植到 WinCC flexible 的项目中使用。WinCC flexible 具有以下特点：

1）功能块库：可自定义及重复使用各种功能块，并可集中更改。

2）动态面板。

3）组态智能工具，例如画面向导、运动对象及方便地组态批量数据（变量编辑器等）。

4）使用用户 ID 或密码进行访问保护。

5）配方管理。

6）报表系统。

7）可打开 ProTool 组态。

8）个性化扩展选项：包括 Sm@rt 客户端/服务器选件、网络服务和诊断选件、OPC 服务器通信选件、过程诊断选件以及操作员操作行为和组态记录/跟踪选件等。

SIMATIC TP177A 具有易于使用、灵活性好的特点，组态软件为 WinCC flexible，通信方式有 PPI/MPI/PROFIBUS 等，本例采用 PPI 通信协议。

3. PLC 程序设计

（1）PLC 变量地址分配表

PLC 变量地址分配表见表 5-13。

表 5-13　PLC 的变量地址分配表

PLC 的地址	功　能
VB0~VB396	USS 指令库存储区
VD440	变频器频率设定
M0.0	HMI 启动按钮
M0.1	HMI 停止按钮

（2）程序的编写

编写 PLC 程序时，采用 USS 指令库提供的通信指令，如图 5-35 所示。

程序编写时一定要注意给 USS 指令分配库存储区，否则编译出现错误，操作步骤如图 5-36 所示。

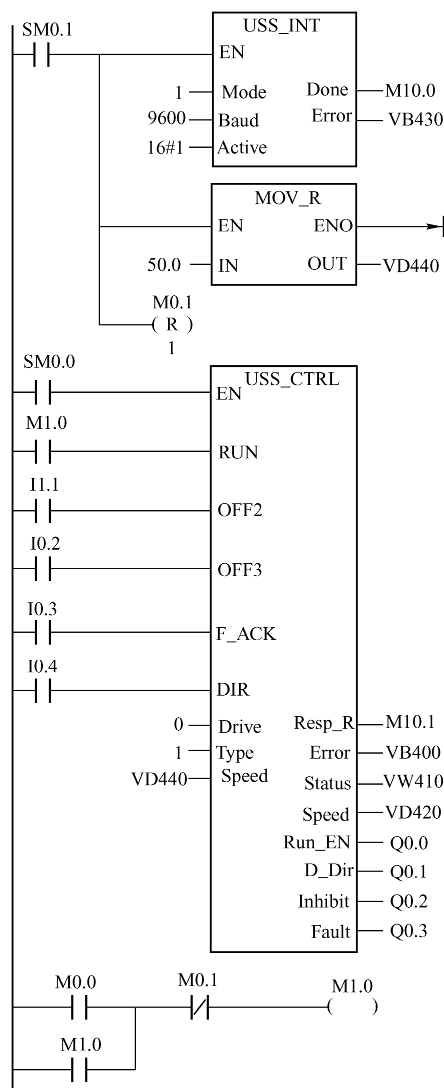


图 5-35　USS 通信程序

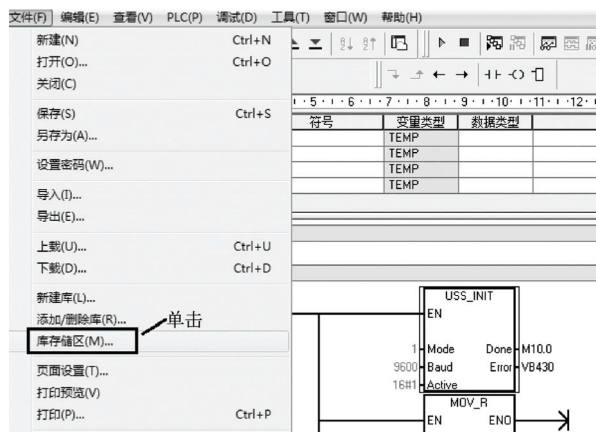


图 5-36　库存储区目录界面

出现如图 5-37 所示界面。

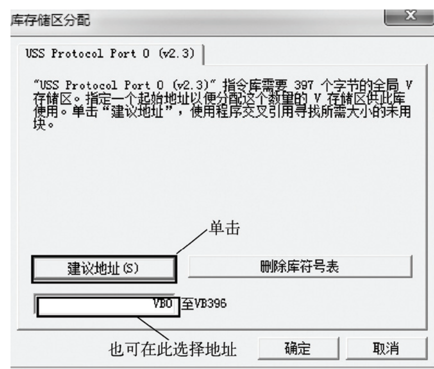


图 5-37　库存储区地址分配

4. HMI 界面设计

（1）创建工程

打开 WinCC flexible 软件，弹出界面如图 5-38 所示，单击“创建一个空项目”，弹出界面如图 5-39 所示。

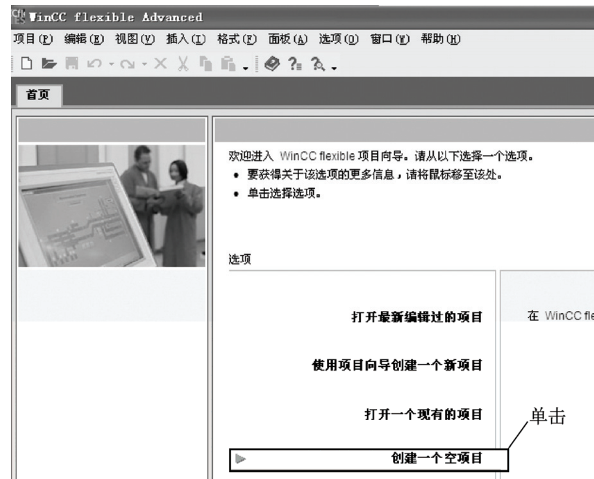


图 5-38　创建工程界面



图 5-39　触摸屏型号选择

（2）设备选择

在图 5-39 中，进入“Panels→170”，选择 HMI 型号“TP177A 6”。

（3）建立连接设备

如图 5-40 所示界面，在“名称”一览输入 HMI 需要连接的设备，如图 5-41 所示。

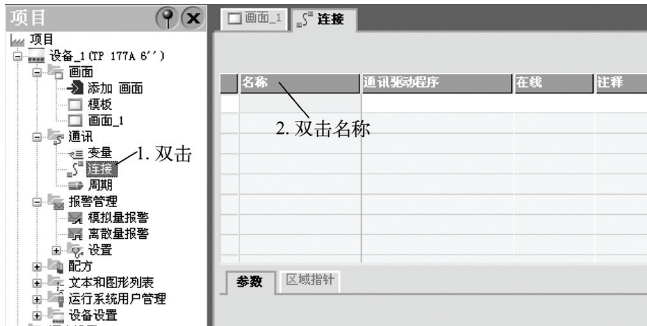


图 5-40　建立连接设备

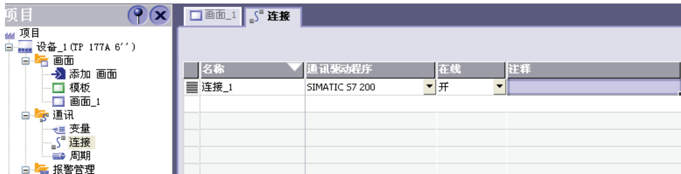


图 5-41　选择“通信驱动程序”

（4）HMI 与 PLC 之间的通信设置

如图 5-42 所示，设置 HMI 与 PLC 通信参数。



图 5-42　HMI 与 PLC 之间的通信设置界面

（5）建立 HMI 控制变量

如图 5-43 所示，建立 HMI 与 PLC 之间的通信变量。



图 5-43　HMI 变量的建立

（6）建立主界面

设计并制作 HMI 操作主界面如图 5-44 所示。

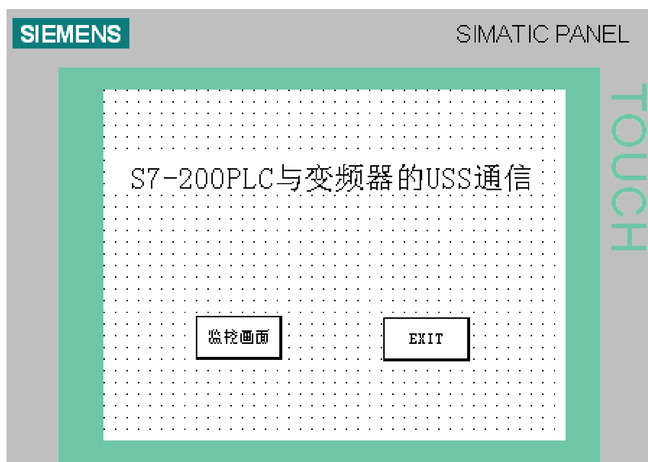


图 5-44 主界面的设计

（7）建立监控界面

设计并制作系统监控界面如图 5-45 所示。

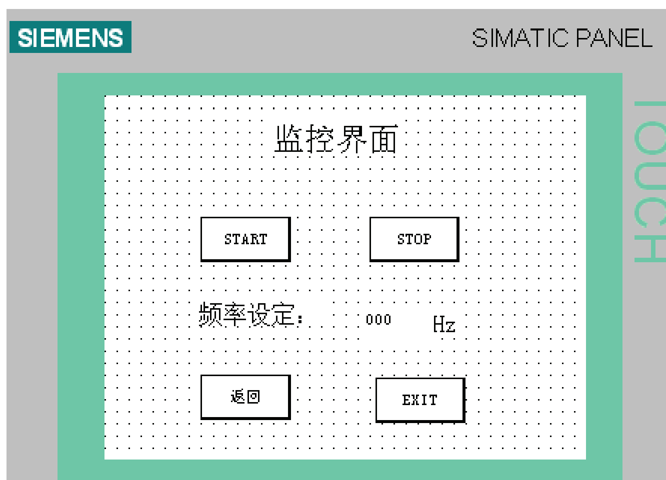


图 5-45 系统监控界面的设计

5. 系统联调

1）使用 PC/PPI 电缆将 PLC 程序和 HMI 程序分别下载到 PLC 和 HMI 屏中。

2）连接 PLC、变频器、HMI。

3）检查并确认变频器进行 USS 通信的参数设置情况。

4）将 PLC 调整为“RUN”运行状态。

5）操作 HMI 监控界面的启动（START）、停止（STOP）按钮，观察变频器运行情况；修改 HMI 监控界面的频率设定值，观察变频器运行时频率的变化情况。

第二篇　S7-300 PLC 基础

第 6 章　S7-300 PLC 的硬件与编程资源

6.1　S7-300 PLC 系统简介

S7 系列 PLC 是德国西门子公司自 1995 年起在 S5 系列 PLC 基础上陆续推出的性价比较高的 PLC 系统。在西门子 S7 家族中，S7-200 PLC 主要用于低端或小型系统，S7-300 PLC 则用于中低端或大中型系统。

S7-300 是模块化的通用型 PLC，各种模块之间可进行广泛的组合和扩展，适用于中等规模性能的控制要求。

S7-300 PLC 系统的各种模块相互独立并安装在固定的机架（导轨）上，用户可根据系统的具体情况选择适合的模块，使控制系统设计更加灵活，以适应不同的应用需求，而且维修时模块更换十分方便；当系统规模扩大和功能复杂时，可以增加模块或机架，对 PLC 系统进行扩展。

S7-300 的 CPU 模块集成了过程控制功能，不需要附加任何硬件、软件和编程，就可以建立一个多点接口（Multi-Point Interface，MPI）网络，该协议是西门子内部协议，是一种通信速率要求不高、通信数据量不大时的简单经济的通信方式，多用于连接上位机和少量 PLC 之间的近距离通信。

新型 S7-300 CPU 没有内置的装载存储器，必须使用 MMC 存储卡作为其装载存储器，用以保存用户数据。当 CPU 断电后会自动将工作存储器中的数据复制到 MMC 中，不需要后备电池就可以长时间保持动态数据，这使得 S7-300 成为完全不需要维护的控制设备，而且用户可以根据程序大小单独购买 MMC 卡，内存范围为 64KB ~ 8MB。

S7-300 PLC 有 350 多条指令，包括位指令、比较指令、定时指令、计数指令、整数和浮点数运算指令等，可用于实现复杂控制功能。CPU 的集成系统提供了中断处理和诊断信息等系统功能，由于它们是集成在 CPU 的操作系统中，因此节省了很多 RAM 空间。

S7-300 系统由于简单实用的分散式结构和多界面网络能力，应用十分灵活；且其产品设计紧凑，在空间有限的场合应用更为出色。

6.2　S7-300 PLC 系统组成

由于 SIMATIC S7-300 PLC 是模块化结构设计，因此可通过 U 型总线把电源、CPU 及其他模块紧密固定在标准导轨上。

如图 6-1 所示，S7-300 PLC 由多种模块部件组成，各模块安装在 DIN 标准导轨上，并用螺钉固定。这种结构形式既可靠，又能满足电磁兼容要求。电源模块（PS）、CPU 模块、接

口模块（IM）、信号模块（SM）、功能模块（FM）等都是独立的模块，每个模块的背板都有一个总线连接器，以便插接其他模块背板，并使背板总线连成一体。

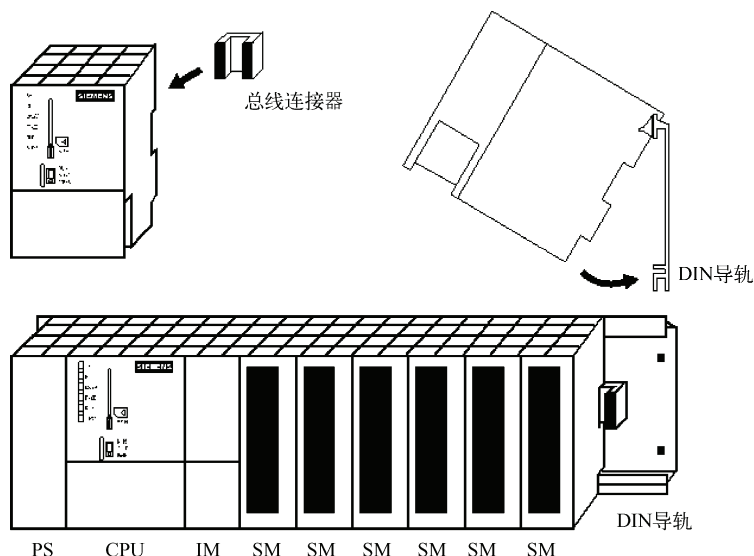


图 6-1　S7-300 的安装

在一个机架上最多可安排不包括 CPU 模块和电源模块的 8 个模块，包括信号模块、功能模块或通信处理器模块，导轨可水平或垂直安装。如果系统任务需要的模块超过 8 块，则可以通过增加扩展机架来增加挂接的模块数量。

电源模块总是安装在机架的最左边，CPU 模块紧靠电源模块。CPU 的右边是可以选择的接口模块，如果系统只用主机架而没有扩展机架时可以不选择 IM 接口模块。除了带 CPU 的中央机架（CR），最多还可以增加 3 个扩展机架（ER）；每个机架可以插接 8 个模块， 4 个机架最多可以安装 32 个模块。

机架的最左边是 1 号槽，最右边是 11 号槽，电源模块总是在 1 号槽的位置。中央机架（0 号机架）的 2 号槽上是 CPU 模块，3 号槽是接口模块。这 3 个槽号被固定占用，信号模块、功能模块或通信模块等使用 4~11 号槽。

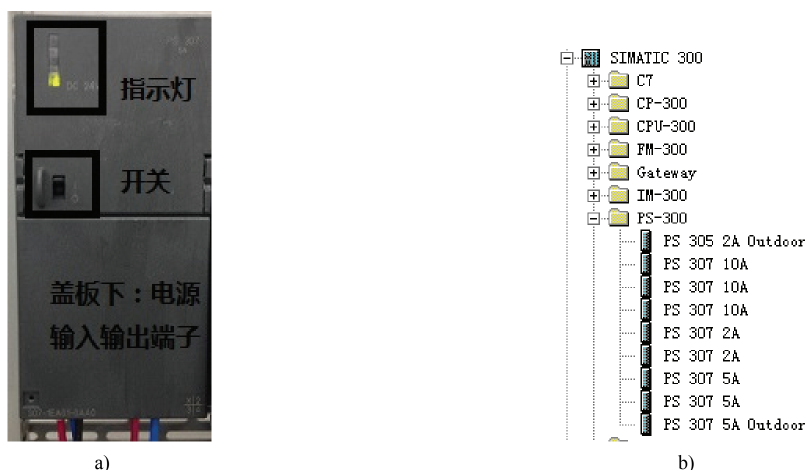
实际上这些都是逻辑位置，系统可以没有电源模块而使用外部开关电源，而接口模块基本上也很少使用。

6.2.1　主要模块介绍

1. 电源模块（PS）

电源模块外部结构如图 6-2a 所示，用于将 AC120V/230V 电源转换为 24V 直流电源，供 CPU 模块和 I/O 模块使用。电源模块的额定输出电流有 2A、5A 和 10A 三种型号，输出电压是隔离的并具有短路保护功能，不带负载时输出稳定。过载时模块上的 LED 闪烁，S7-300 模块使用的电源由 S7-300 背板总线提供，一些模块还需从外部负载电源供电。如果有其他电源且符合设计要求，则也可以不配置 S7-300 的电源模块。

使用 S7-300 PLC 编程软件进行硬件组态时，可选电源类型如图 6-2b 所示。



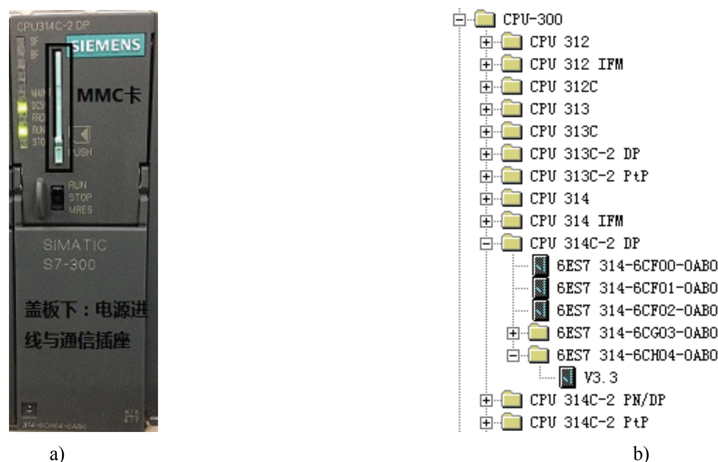
a) b)

图 6-2　电源模块

a)外部结构　b)电源类型

2. 中央处理单元（CPU）

CPU 模块外部结构如图 6-3a 所示，CPU 是 PLC 系统的运算控制核心，用于存储和处理用户程序，控制集中式 I/O 和分布式 I/O。各种 CPU 有不同的性能，有的 CPU 集成有数字量和模拟量输入/输出点，有的 CPU 集成有 PROFIBUS-DP/PROFINET 等通信接口。CPU 前面板上有状态故障指示灯、微存储卡插槽、模式选择开关、通信接口以及 24V 电源端子。



a) b)

图 6-3　CPU 模块

a) 外部结构　b) CPU 类型

CPU 存储器可以分为三个区域：工作存储器（RAM）、系统存储区、装载存储器（位于 MMC 卡中）。

RAM 集成在 CPU 中，具有保护功能，不能被扩展，用于运行程序指令，并处理用户程序数据。

系统存储器集成在 CPU 中，不能被扩展，包括标志位、定时器及计数器的地址区，I/O 的过程映像以及局域数据。

由于 CPU31xC 没有安装集成的装载存储器，在使用 CPU 时必须插入一个 SIMATIC 微型存储卡（MMC），用于保存程序指令块和数据块以及系统数据，也可以将项目的组态数据保存在 MMC 卡中，MMC 卡外观如图 6-4 所示。MMC 卡的读/写直接在 CPU 内进行，不需要专用的编程设备，CPU 与 MMC 卡是分开订货的。



图 6-4　MMC 卡外观

如果在写访问过程中拆下 MMC 卡，卡中的数据会被破坏，在这种情况下必须将 MMC 插入 CPU 中并删除它，或在 CPU 中格式化存储卡；只有在断电状态下或 CPU 处于“STOP”状态时，才能取下存储卡。

程序只能在 RAM 和系统存储器中运行。使用 S7-300 PLC 编程软件进行硬件组态时，可选 CPU 类型很多，其中每一种类型又分不同的货号和版本号，例如图 6-3b 所示的 CPU314C-2DP，如果需要选择的 CPU 货号或版本号不存在，则需要安装该 CPU 的 GSD 文件。

3. 信号模块（SM）

信号模块是数字量输入/输出（简称为 DI/DO）模块和模拟量输入/输出（简称为 AI/AO）模块的总称，用于将外部不同的过程信号与 S7-300 输入/输出相匹配，用户可根据现场输入/输出设备选择不同用途的 I/O 模块。使用 S7-300 PLC 编程软件进行硬件组态时，SM 模块分类如图 6-5 所示，每种类型下又提供了许多不同点数、不同货号或不同电压的模块供用户选择。

信号模块面板上的 LED 用来显示各数字量输入/输出点的信号状态。附件有总线连接器与前连接器。

每个模块上有一个背板总线连接器，用于模块之间的连接。

现场的过程信号连接到前连接器的端子上，S7-300 的输入/输出模块的外部接线接在插入式的前连接器的端子上，前连接器插在前盖后面的凹槽内，不需断开前连接器上的外部连线，就可以迅速更换模块。

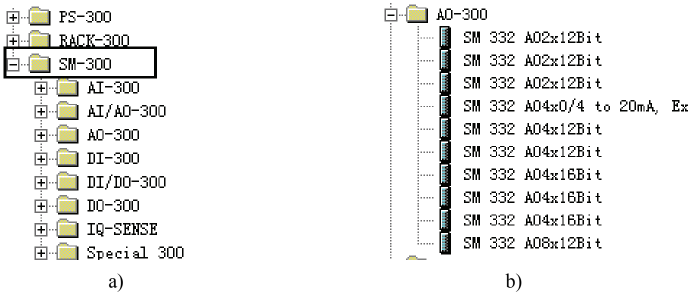


图 6-5　SM 模块分类

a) 信号模块分类　b) 模拟量输出模块分类

4. 功能模块（FM）

功能模块是智能的信号处理模块，不占用 CPU 的资源，可对来自现场设备的信号进行控制和处理，并将信息传送给 CPU；它们负责处理那些 CPU 通常无法以规定的速度完成的任务，以及对实时性和存储容量要求很高的控制任务，如用于完成高速计数的 FM350 计数模块、用于完成定位功能的 FM351 定位模块、用于完成 PID 控制功能的 FM355 闭环控制模块。STEP 7 V5.5 版本硬件组态时，提供的 FM 模块类型如图 6-6 所示。

5. 通信处理器（CP）

通信处理器用于 PLC 之间、PLC 与计算机和其他智能设备之间的通信，可以减轻 CPU 处理通信的负担，并减少用户对通信的编程工作，如 CP343-1 IT 为工业以太网处理器，CP342-5 为 PROFIBUS-DP 通信处理器，CP340 为点到点通信处理器。STEP 7 V5.5 版本硬件组态时，提供的 CP 模块类型如图 6-7 所示。

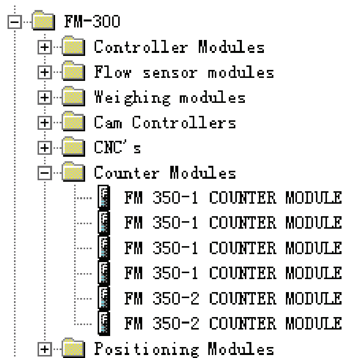


图 6-6　FM 模块类型

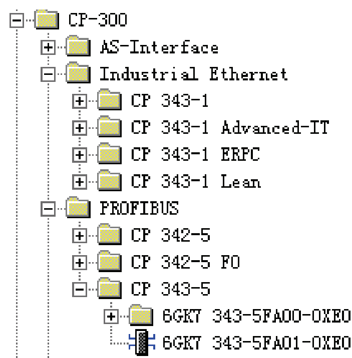


图 6-7　CP 模块类型

6. 接口模块（IM）

在 S7-300 系统中，除 CPU312/312C 外，当 PLC 的 I/O 点数超过 256 点或安装的模块数超过 8 个时，需要在基本机架的基础上通过扩展机架进行扩展，最多可连接 3 个扩展机架。

接口模块必须安装到特定的插槽中，用于多机架配置时连接主机架（CR）和扩展机架（ER）之间的总线。STEP 7 V5.5 版本硬件组态时，提供的 IM 模块类型如图 6-8 所示，接口模块 IM360 是发送模块，安装在主机架的 3 号插槽中，IM361 是接收模块，安装在扩展机架上，IM361 还可以连接其他 IM361 模块，即多级扩展时可以通过 IM361 依次串联连接，但受到 S7-300 最大扩展级的限制；IM365 只能用于一级扩展，即仅能连接一个扩展机架，扩展时在基本机架与扩展机架上均应安装一个 IM365 模块。

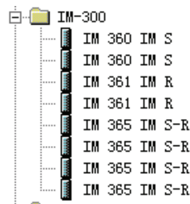


图 6-8　IM 模块类型

7. 导轨

导轨是 S7-300 的机架，通常是铝质材料，用来固定和安装 S7-300 的各种模块。导轨没有背板总线，CPU 与模块是通过 U 型连接器连接的。

S7-300 有两种型式的导轨：

1）2m 导轨，导轨中没有固定螺孔和接地导轨螺栓，在安装时需要 4 个用于固定螺钉的孔和 1 个用于保护导线的螺栓孔并钻孔（例如 M6 螺钉，孔直径为 6.5 + 0.2mm）。

2）简单易用，共有四种标准长度，分别为 160mm/482.6mm/530mm/830mm，提供 4 个固定螺钉孔、1 个接地导线螺栓。

根据需要可以截短导轨，以便用于特殊长度的子网组态。

6.2.2　CPU 的操作模式

1. CPU 的模式选择

如图 6-9 所示的 CPU 模块，其模式开关有三种状态选择，各位置的意义如下：

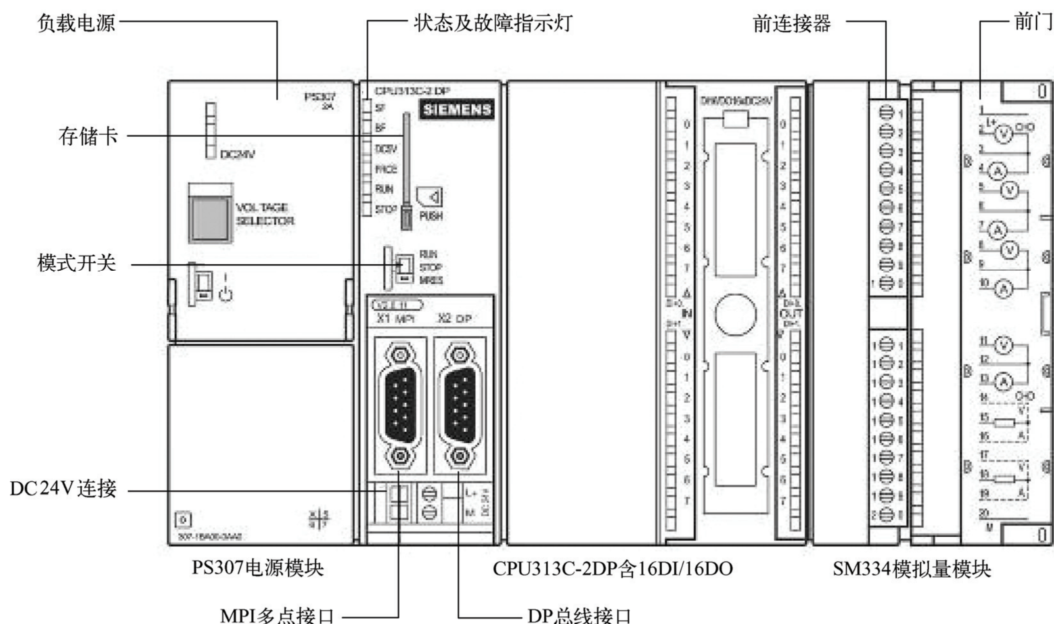


图 6-9　S7-300 CPU 结构

1）RUN（运行）位置：CPU 执行用户程序，可以通过编程软件读出用户程序，但是不能修改用户程序。

2）STOP（停止）位置：不执行用户程序，通过编程软件可以读出和修改用户程序。

3）MRES（清除存储器）：MRES 位置不能保持，在这个位置松手时开关将自动返回 STOP 位置。将模式选择开关从 STOP 状态扳到 MRES 位置时，可以复位存储器，使 CPU 回到初始状态。工作存储器、RAM 装载存储器中的用户程序和地址区被清除，全部存储器位、定时器、计数器和数据块均被复位为零，包括所有保持功能的数据。CPU 检测硬件、初始化硬件和系统程序的参数、系统参数、CPU 和模块的参数被恢复为默认设置，MPI 的参数被保留。如果有 MMC 卡，CPU 在复位后将它里面的用户程序和系统参数复制到工作存储区。

CPU 在运行中如果有错误，则模块面板的状态及故障指示灯会显示，指示灯有 6 个，用于显示运行状态和故障，表 6-1 列出了指示灯在系统稳定状态时灯亮的含义。

表 6-1　CPU 操作面板指示灯含义

发光二极管	含　　义	说　　明
SF（红色）	系统错误/故障	下列事件引起灯亮： 1）硬件故障；2）固件出错；3）编程出错 4）参数设置出错；5）算数运算出错 6）定时器出错 7）存储器卡故障（只在 CPU313 和 CPU314 上） 8）电池故障或电源接通时无后备电池（只在 CPU313 和 CPU314 上） 9）输入/输出的故障或错误（只对外部 I/O） 可用编程装置读出诊断缓冲器中的内容，以确定错误/故障的真正原因
BATF（红色，只在 CPU313 和 314 上有）	电池故障	如果电池失效或未装入，则灯亮
DC5V（绿色）	用于 CPU 和 S7-300 的 DC5V 电源	如果内部的 DC5V 电源正常，则灯亮
FRCE（黄色）	强制状态	强制作业有效
RUN（绿色）	系统运行	当模块处于运行状态时，指示灯亮
STOP（黄色）	系统停止	当模块处于停止状态时，指示灯亮

2．CPU 操作模式

S7-300 CPU 通常有 4 种操作模式：STOP（停机）、RUN（运行）、HOLD（保持）和 STARTUP（启动）。在所有的模式中，都可以通过 MPI 接口与其他设备通信。

1）STOP 模式：模式选择开关在 STOP 位置时，CPU 模块上电后自动进入 STOP 模式，在该模式下不执行用户程序，可以接收全局数据和检查系统。

2）RUN 模式：执行用户程序，刷新输入和输出，处理中断和故障信息服务。

3）STARTUP 模式：可以用模式选择开关或编程软件 STEP 7 启动 CPU。如果模式选择开关在 RUN 位置，通电时自动进入启动模式。

4）HOLD 模式：在 STARTUP 和 RUN 模式执行程序时遇到调试用的断点，用户程序的执行被挂起（暂停），定时器被冻结。

3．S7-300 CPU 的分类

S7-300 CPU 有 20 多种不同的类型，各种 CPU 按性能等级划分，可以涵盖各种应用范围。S7-300 的各类 CPU 都有详细的性能介绍，具体参数可查阅《SIMATIC S7-300 CPU31xC 和 CPU31x：技术规范》或其他相关资料，CPU 的主要性能包括 5 个方面：I/O 扩展能力、指令执行速度、工作内存容量、通信能力及 CPU 上的集成功能。

S7-300 提供了不同性能档次的 CPU 模块。例如标准型 CPU 提供范围广泛的基本功能，包含指令执行、I/O 读写、通过 MPI 和 CP 模块通信等；紧凑型 CPU 本机集成 I/O，并带有高速计数、频率测量、定位和 PID 调节等技术功能；部分 CPU 还集成了点到点或 PROFIBUS 通信接口。S7-300 的 CPU 模块可分为以下几种类型：

1）标准型 CPU：CPU 312、CPU 314、CPU 315-2DP、CPU 315-2PN/DP、CPU 317-2DP、CPU 317-2PN/DP 和 CPU 319-3PN/DP。

2）紧凑型 CPU：CPU 312C、CPU 313C、CPU 313C-PtP、CPU 313C-2DP、CPU 314C-PtP 和 CPU 314C-2DP。

3）技术功能型 CPU：CPU 315T-2DP 和 CPU 317 T -2DP。

4）故障安全型 CPU：CPU 315F-2DP、CPU 315F-2PN/DP、CPU 317F-2DP 和 CPU 317F-2PN/DP。

5）SIPLUS 户外型 CPU：可以在环境温度-25～+70ºC 和有害的气体环境下运行。

6.3 STEP 7 编程软件

S7-300 系列 PLC 的编程软件是 STEP 7，用文件块的形式管理用户编写的程序及程序运行所需的数据，并组成结构化的用户程序。这样构成的 PLC 程序组织明确、结构清晰、易于修改。

STEP 7 编程软件用于 SIMATIC S7、C7、M7 和基于 PC 的 WinAC，并为它们提供组态、编程和监控等服务。主要完成以下功能：

1）SIMATIC 管理器。用于集中管理所有工具以及自动化项目数据。

2）组态硬件。即在机架中放置模块，为模块分配地址和设置模块的参数。

3）程序编辑器。使用编程语言编写用户程序。

4）符号编辑器。用于管理全局变量。

5）组态通信连接、定义通信伙伴和连接特性。

6）下载和上传用户程序、调试用户程序、启动、维护、文件建档、运行和诊断等功能。

STEP 7 的所有功能均有在线帮助，用鼠标打开或选中某一对象，按 F1 键（或 F1+Fn）可以得到该对象的在线帮助。

6.3.1 STEP 7 的安装

1）执行 STEP 7 V5.5 安装盘根目录下的 Setup.exe。如果出现如图 6-10 所示的错误信息，则需要重新启动计算机；如果仍安装不上，就要进入注册表“regedit”路径“HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager”，检查是否出现值“PendingFileNameOperations”，如果存在，将其删除，重新安装程序。

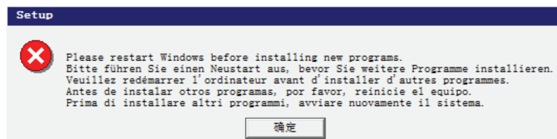


图 6-10 安装 Step 7 的出错信息

2）安装程序进入后，选择需要安装的项目，如图 6-11 所示，建议选择全部安装。

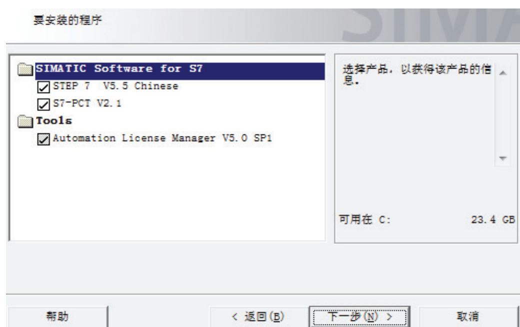


图 6-11 选择安装项目

3）按提示逐步安装所有项目。根据计算机性能的不同，需要大约数十分钟的时间，在安装过程中要求选择安装模式，通常选择 Typical，如图 6–12 所示；语言安装选择后，界面如图 6–13 所示；继续安装界面如图 6–14 所示。



图 6–12 选择安装模式



图 6–13 语言选择后出现界面

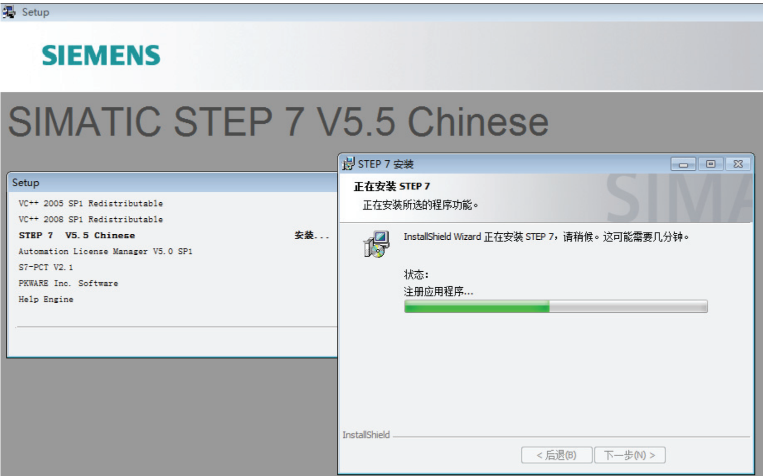


图 6–14 继续安装界面

4）进行通信接口设置，如图 6-15 所示。在左边的“选择：”一列中选择 PC Adapter，单击“安装”按钮。如果系统使用 CP5611 卡通信，则选中“CP5611”→“安装”。单击“关闭”按钮，出现界面如图 6-16 所示，按下“确定”按钮，弹出界面如图 6-17 所示。



图 6-15　通信接口设置

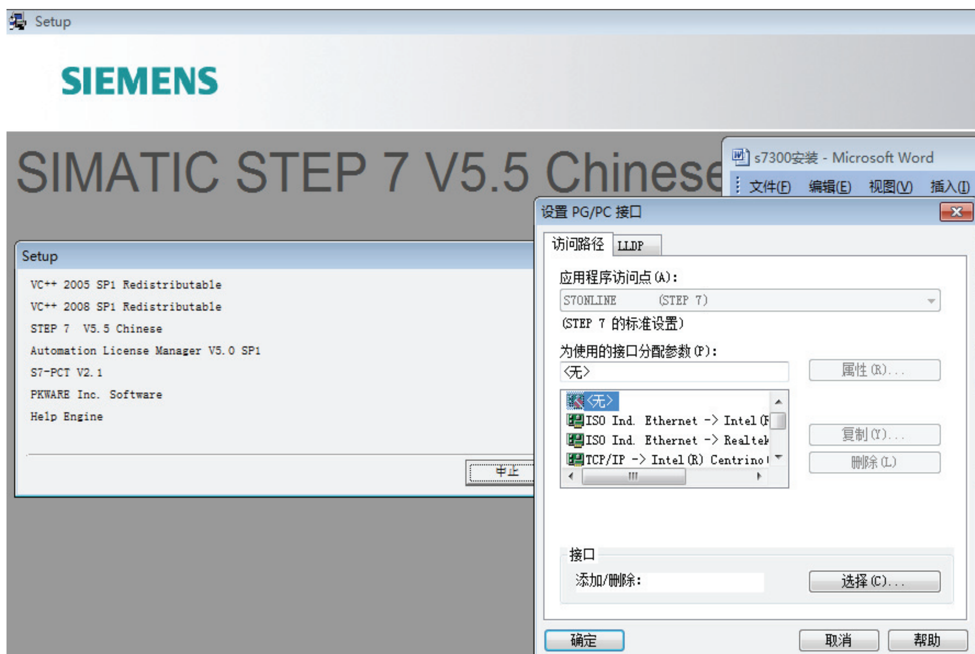


图 6-16　设置 PG/PC 接口

PC Adapter 支持 MPI 和 PROFIBUS 两种通信方式，如 S7-300 使用 MPI 通信方式时，可用 MPI 电缆将计算机 COM 口与 S7-300 的 MPI 通信口连接使用。

如果计算机装有 CP5611 卡，则可以通过此卡将计算机连接到 MPI 或 PROFIBUS 网络。如果采用这种方式，还需要安装 CP5611 卡驱动程序。

5）如图 6-17 所示，安装完成后，需要重新启动计算机。



图 6-17　安装完毕后重新启动计算机

6.3.2　STEP 7 的硬件接口

在 PC 上使用 STEP 7，必须配置 PC/MPI 通信适配器或通信卡，将计算机接入 MPI 网络或 PROFIBUS 网络，以便下载和上传用户程序及组态数据。

PC/MPI 适配器用于连接安装了 STEP 7 的计算机的 RS-232C 接口和 PLC 的 MPI 接口，如图 6-18 所示。计算机一侧的通信速率为 19.2kbit/s 或 38.4kbit/s，PLC 一侧的通信速率为 19.2kbit/s~1.5Mbit/s。



图 6-18　PC/MPI 适配器

使用计算机的通信卡 CP 5611（PCI 卡）、CP 5511 或 CP 5512（PCMCIA 卡），可以将计算机连接到 MPI 或 PROFIBUS 网络，通过网络实现计算机与 PLC 的通信。

也可以使用计算机工业以太网通信卡 CP 1512（PCMClA 卡）或 CP 1612（PCI 卡），通过工业以太网实现计算机与 PLC 的通信。

在计算机上安装好 STEP 7 后，在管理器中执行菜单命令“选项”→“设置 PG/PC 接口”，打开“设置 PG/PC 接口”对话框，如图 6-19 所示，单击“确定”按钮后，弹出图 6-20 所示的提示界面，单击“确定”按钮后，接口协议改变完成。

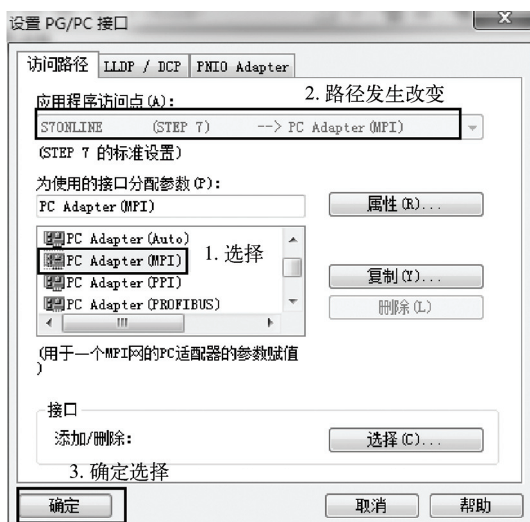


图 6-19 接口协议设置

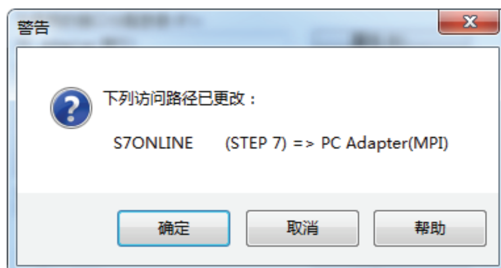


图 6-20 提示界面

在 STEP 7 Manager 界面右下方会显示当前接口协议，如图 6-21 所示。

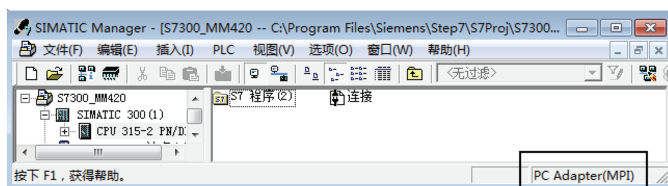


图 6-21 管理器界面

如果在“设置 PG/PC 接口”界面没有要选择的接口协议，则按照图 6-22 所示的步骤，完成安装接口功能，再按照图 6-19 和图 6-20 所示的步骤选择、操作。



图 6-22 添加接口协议

6.3.3 STEP 7 的授权

授权是使用 STEP 7 编程软件的“钥匙”，只有安装了授权，才可以正常使用 STEP 7 软件。没有授权也可以使用 STEP 7，以便用户熟悉接口和功能，但是在使用时每隔一段时间将会搜索授权，提醒使用者安装授权。在购买软件时会附带一张包含授权的软盘，用户可在安装过程中将授权从软盘转移到硬盘上，也可以在安装完毕后的任何时间内使用授权管理器完成转移，有关授权内容请看相关技术文件。

6.3.4 创建一个工程

双击软件快捷键“SIMATIC Manager”，打开软件。单击“File”→“New…”，新建一个工程项目，例如 Test，如图 6-23 所示。

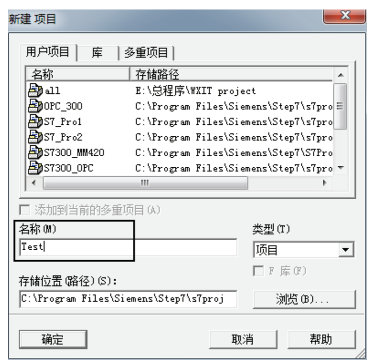


图 6-23 新建工程

单击“确定”按钮，生成如图 6-24 的新工程 Test。

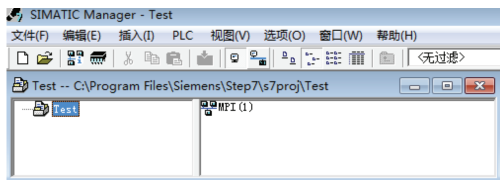


图 6-24 新工程

在图 6-24 界面上建立 S7-300 站。鼠标右键单击工程名 Test，单击“插入新对象”→“SIMATIC 300 站点”，完成界面如图 6-25 和图 6-26 所示。

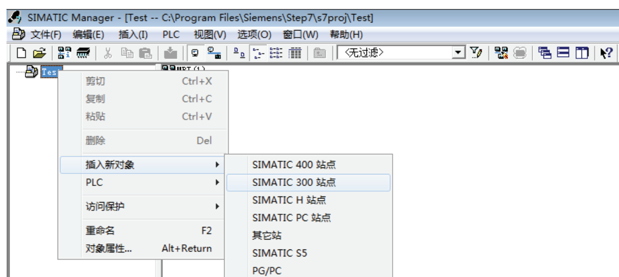


图 6-25 建立 S7-300 站

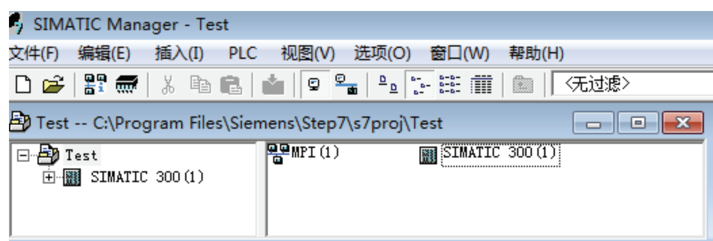


图 6-26　添加站点完成

如图 6-27 所示，单击并展开 Test 文件夹，双击“硬件”，即可进入硬件组态环境。如图 6-28 所示。

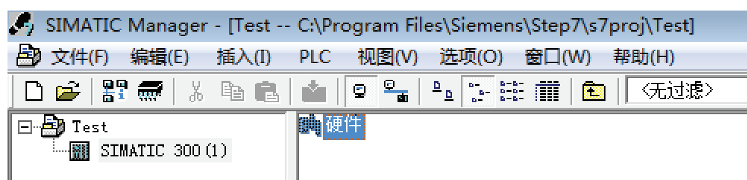


图 6-27　双击“硬件”开始硬件组态

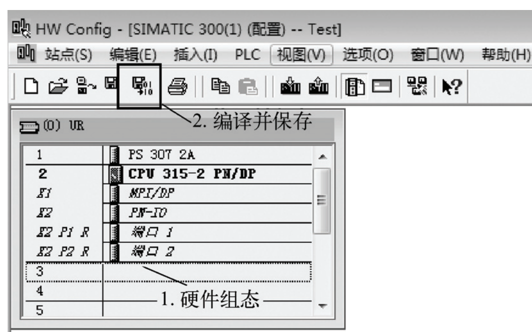


图 6-28　硬件组态界面

硬件组态完毕后，管理器界面 SIMATIC 300(1) 中出现 CPU 型号，如图 6-29 所示。展开至 Blocks（块），出现 OB1，打开 OB1 即可进行程序编写。

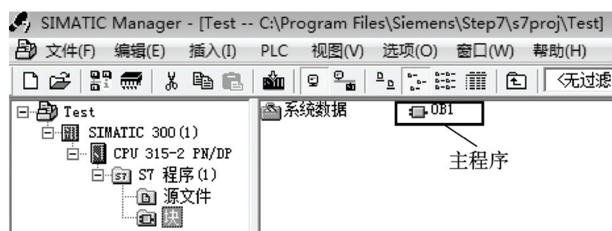


图 6-29　在 Blocks 中进行编程

图 6-29 清晰地显示了项目的分层结构。在项目中，数据在分层结构中以对象的形式保存，左边窗口内的树（Tree）显示项目的结构。第一层为项目，第二层为站（Station），站是组态

硬件的起点。“S7 程序（1）”文件夹是编写程序的起点，所有的软件均存放在该文件夹中。用鼠标选中图中某一层的对象，在管理器右边的工作区将显示所选文件内的对象和下一级的文件夹。双击工作区中的图标，可以打开并编辑对象。

Blocks（块）对象包含程序块（Blocks）、用户定义的数据类型（UDT）、系统数据（System data）和调试程序用的变量表（VAT）。程序块包括逻辑块（OB、FB、FC）和数据块（DB），需要把它们下载到 CPU 中，用于执行自动控制任务，符号表、变量表和 UDT 不用下载到 CPU 中。生成项目时会在块文件夹中自动生成一个空的组织块 OB1，OB1 是用户程序的接口，也是系统必需的程序块。

用户生成的变量表（VAT）在调试用户程序时用于监视和修改变量。系统数据块（SDB）中的系统数据含有系统组态和系统参数的信息，它是用户进行硬件组态时提供的数据自动生成的。图 6-30 是编辑完成的项目结构。

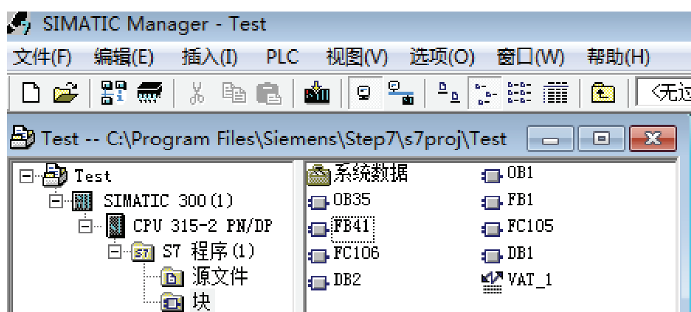


图 6-30 编辑完成的项目结构

如果硬件组态或程序不能下载，需要检查通信接口导线连接是否正确以及通信设置是否正确。如果通信电缆采用 PC Adapter(MPI)，通信设置步骤见 6.3.2 （STEP 7 的硬件接口）。如果找不到 PC Adapter 协议，则按照图 6-31 所示安装通信接口，回到图 6-19 “设置 PC/PG 接口”界面，单击“属性”按钮，设置或添加 PC Adapter（MPI）属性，如图 6-32 所示。“属性”页面中选择“本地连接”，选择 COM1、19200 参数，注意一般连接到计算机的串行口 1，其他参数默认即可，如图 6-33 所示。

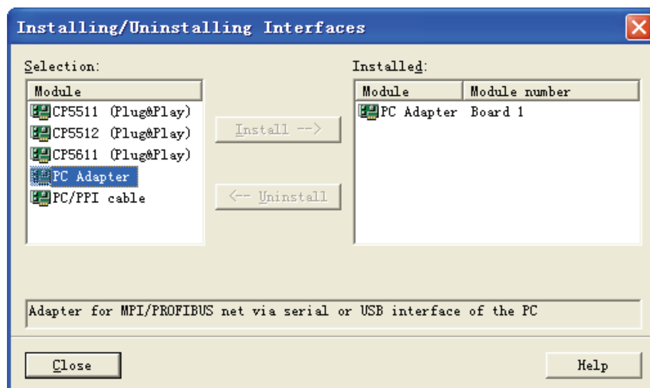


图 6-31 安装/卸载通信接口

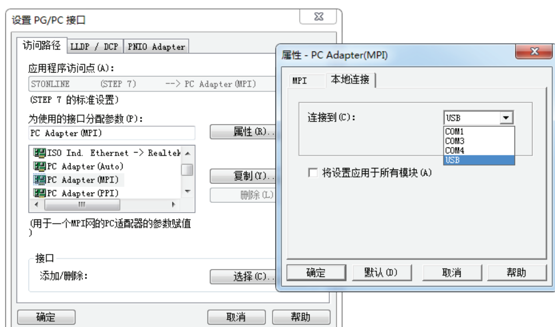


图 6-32　MPI 电缆连接参数设置



图 6-33　串口参数

6.3.5　STEP 7 的程序块

在 STEP 7 软件中，PLC 中的程序分为操作程序和用户程序。操作程序用于实现与特定的控制任务无关的功能，处理 PLC 的启动、刷新过程映像输入/输出表、调用用户程序、处理中断和错误、管理存储区和处理通信等；用户程序包含处理用户特定的自动化任务所需要的所有功能。

结构化的用户程序是以“块”形式实现的，显示界面如图 6-30 所示。STEP 7 将用户自己编写的程序数据放置在块中，使程序部件标准化，通过块与块之间的调用，实现用户的控制功能。用户程序中的“块”对象分类如图 6-34 所示，使用情况见表 6-2。

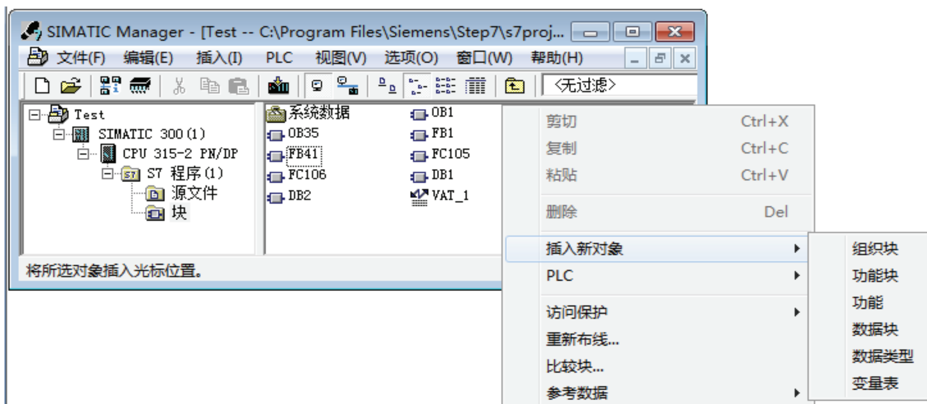


图 6-34　“块”对象

表 6-2　用户程序功能块情况一览表

块　名　称	功 能 概 述
组织块（OB）	系统自动调用，是操作系统与用户程序的接口，决定用户程序的结构
数据块（DB）	存储用户数据及程序的中间变量，为全局变量，最大容量为 32KB。可分为共享数据块（DB）、背景数据块（Instance DB），其中背景数据块是在调用 FB 和 SFB 时用于传递参数的数据块，编译时自动生成数据，不需要用户操作
功能块（FB）	用户编写的包含常用功能的子程序，有专用的背景数据块，用来存储接口数据区和运算的中间结果，其他程序可以直接使用背景数据区中的数据；程序的最大容量为 16KB
功能（FC）	用户编写的程序，没有专用的背景数据块，既可作为子程序用，也可作为函数用，函数中通常带形参。程序的最大容量为 16KB
系统功能块（SFB）	集成在 CPU 模块中，相当于系统提供的可供用户调用的 FB。通过 SFB 调用系统功能，有专用的背景数据块

（续）

块　名　称	功 能 概 述
系统功能（SFC）	集成在 CPU 模块中，相当于系统提供的可供用户调用的 FC。通过 SFC 调用系统功能，无专用的背景数据块
数据类型（UDT）	全局变量，指定程序中数据元素的大小与格式，以便用户使用
变量表（VAT）	用来在程序调试和运行时修改、监视变量内容

由表 6-2 可见，OB、FB、FC、SFB、SFC 均包含部分程序，因此也称作逻辑块。S7 CPU 提供标准系统功能块 SFB、SFC，用户可以直接调用它们，由于它们是操作系统的一部分，因此不需将其作为用户程序下载到 PLC 中；FB、FC 实际上是用户子程序，分为带记忆功能的功能块 FB 和不带记忆功能的功能块 FC；DB 是由用户定义的存储区，用于 FB 或 FC 的数据存取，可以打开或关闭，可以是属于某个 FB 的背景数据块，也可以是通用的全局数据块。

CPU 循环执行操作系统程序，在每次循环中，操作系统程序调用一次主程序 OB1，在任何情况下，OB1 都是必需的。其他大多数 OB 则对应于不同的中断处理程序，如果出现中断事件，CPU 将停止当前正在执行的程序，转去执行中断事件对应的组织块 OB；中断程序执行完以后，再返回到程序中断处继续执行先前正在执行的程序。

在程序设计过程中，可以将一个项目控制任务划分为各个较完整的单独的任务，分别建立和编写与各个任务相对应的逻辑块，块与块之间可以根据控制要求相互调用，被调用的块是 OB 之外的逻辑块。

6.4　简单项目的建立及运行

6.4.1　控制要求

按下启动按钮 SB1（I0.0），PLC 驱动指示灯 LD1（Q0.0）亮。

6.4.2　硬件组态

进入 SIMATIC Manager 界面，按照图 6-23~图 6-27 所示的步骤建立新工程“test”，在 STEP 7 Manager 界面中，双击右侧“硬件”，弹出界面如图 6-35 所示，可进行项目的硬件配置与组态。

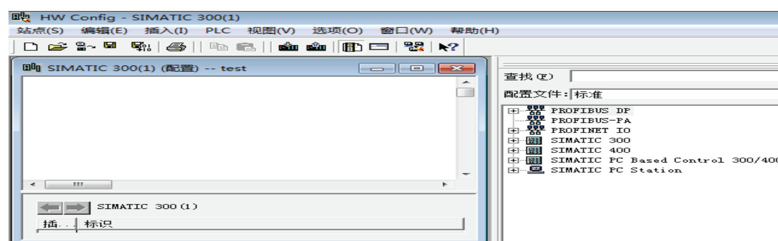


图 6-35　硬件组态界面

配置顺序：“视图”→“目录”→“SIMATTIC 300”菜单，分别进入 RACK-300、PS-300、CPU-300 子菜单，插入机架，在 1 号槽中插入电源模块；在 2 号槽中插入 CPU 模块。图 6-36

所示为电源模块订货号的选择与插入。

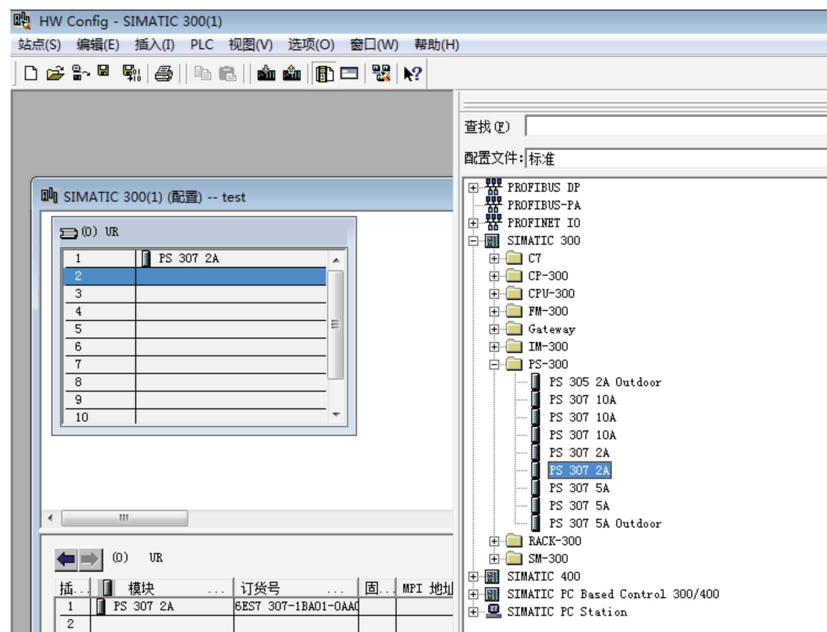


图 6-36　电源模块的选择与组态

由于所选 CPU314C-2DP 模块集成了 24 点数字量输入、16 点数字量输出、5 路模拟量输入、2 路模拟量输出、高速计数通道和定位通道，因此选中 CPU 订货号后，组态界面如图 6-37 所示，系统已自动为输入输出模块分配地址。

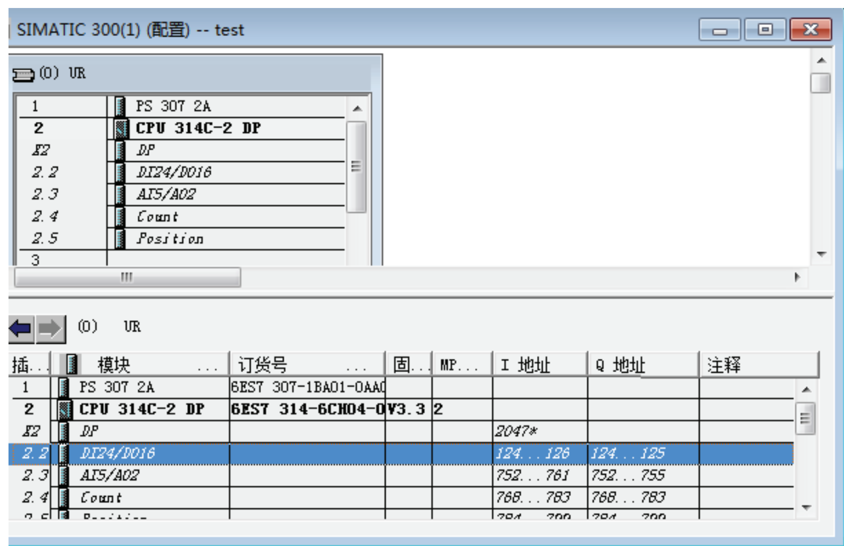


图 6-37　CPU 模块组态

如果要修改输入/输出地址，可双击图 6-37 中的 2.2 行，弹出输入/输出“属性”界面，如图 6-38 所示，取消勾选系统默认的复选框，并填写需要设定的首地址就可以了。



图 6-38　修改输入/输出地址

硬件配置后，单击保存编译快捷键“ ”，然后单击下载快捷键“ ”，将硬件配置下载至 PLC 中，如图 6-39 所示。

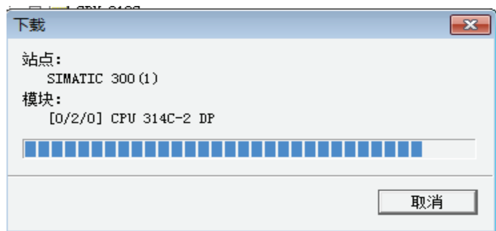
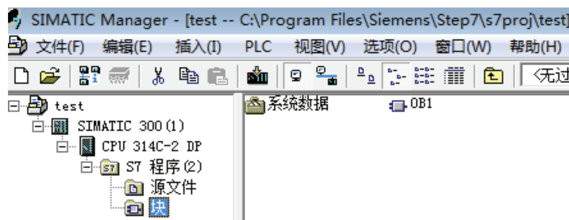


图 6-39　硬件下载过程

硬件下载完毕回到 STEP7 Manager 界面，展开树形菜单，选中“块”，如图 6-40 所示，右侧出现 OB1 块，双击 OB1，弹出界面如图 6-41 所示。



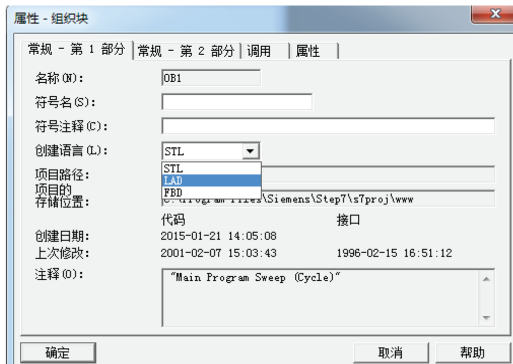


图 6-40　展开项目结构　　　　图 6-41　打开 OB1 并选择编程语言

6.4.3　程序编写

如图 6-41 所示，将编程语言改成梯形图 LAD，单击“确认”按钮，进入 OB1 程序的编写界面，编写梯形图如图 6-42 所示。

程序编写完毕，单击保存快捷键“ ”，并单击下载快捷键“ ”，弹出如图 6-43 所示的界面，单击“是”按钮，程序下载并完成。

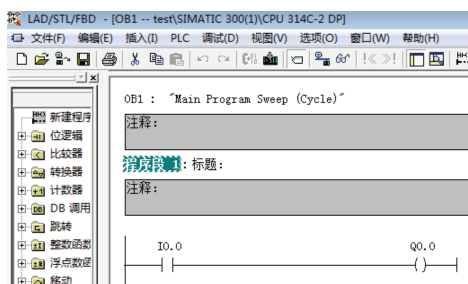


图 6-42 梯形图编写

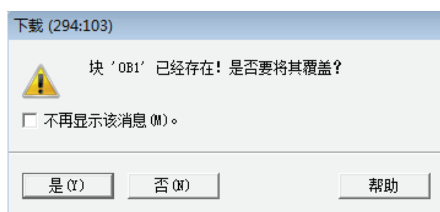


图 6-43 程序下载过程

6.4.4 系统联调

如果将 CPU 模块上的模式开关拨在“RUN”位置，则可以通过软件进行 PLC 状态的远程操作，如图 6-44 所示，进入 PLC “工作模式”，弹出界面如图 6-45 所示。

由图 6-45 可见，如果当前 CPU 为停止状态，则选择“暖启动”按钮，弹出提示界面，选择“是”，CPU 工作模式修改完成，如图 6-46 所示。



图 6-44 选择 PLC 工作模式

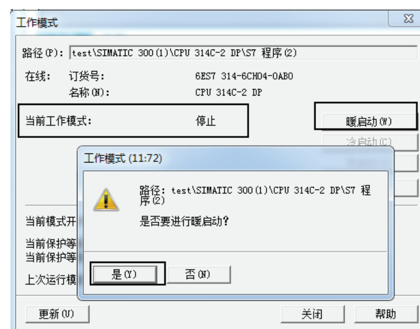


图 6-45 CPU 工作模式切换

在 OB1 界面中，单击快捷键“ ”进行在线监控，监控界面如图 6-47 所示。

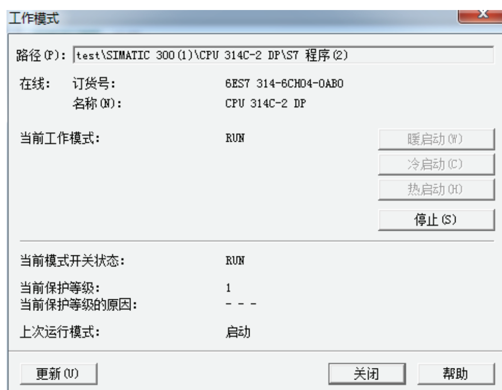


图 6-46 当前工作模式显示

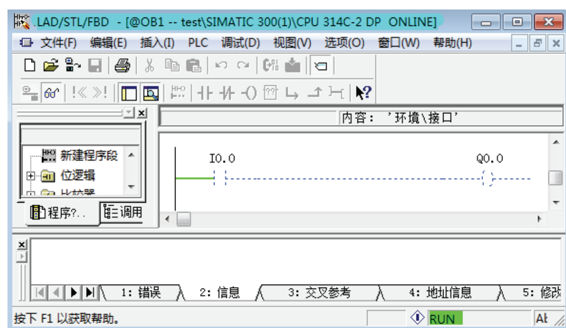


图 6-47 进入程序监控界面

当 I0.0 接通时，Q0.0 得电，如图 6-48 所示，可见在线监控时当线路闭合，则线条颜色

由蓝变绿且由虚线变为实线，表示该线路导通，有能流通过。

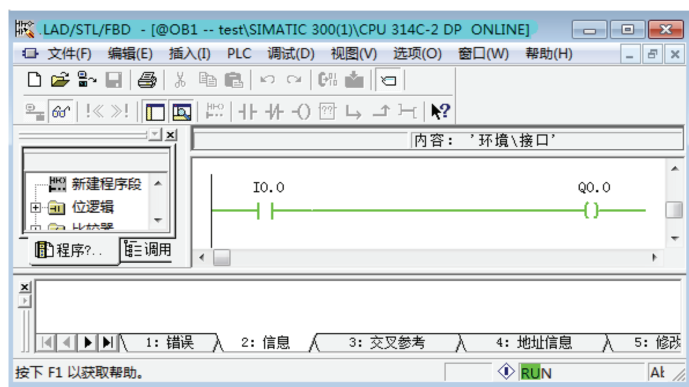


图 6-48　I0.0 闭合，输出 Q0.0 状态为 1

第 7 章　S7-300 PLC 指令系统

7.1　数据及寻址地址

7.1.1　数据块（DB）

数据块（DB）定义为用于存储数据的存储器。用户可在存储器中建立一个或多个数据块，每个数据块可大可小，但 CPU 对数据块数量及数据总量有限制，例如 CPU313C-2DP 最大数据块数量为 1024 个，最大数据块的大小为 64KB。数据块可用来存储用户程序中逻辑块的变量数据，与临时数据不同，当逻辑块执行结束或数据块关闭时，数据块中的数据保持不变。

1．数据块的分类

DB 没有专门的 STEP 7 指令，STEP 7 按数据顺序自动地为 DB 块中的变量分配地址。可分为共享数据块和背景数据块，它们的数据可以被任何一个 OB、FC 或 FB 读写，如图 7-1 所示。

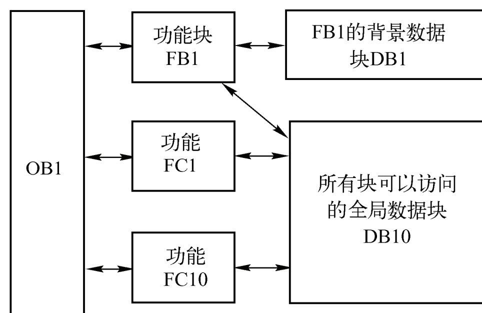


图 7-1　DB 块的应用

共享数据块属于任何逻辑块，它含有生产线或生产设备所需的数值，可以在程序的任何位置直接使用，所有逻辑块（OB、FC、FB）都可以访问共享数据块存储的数据。

背景数据块直接属于某个逻辑块，例如 DB1 专属于 FB1，FB1 的参数和静态变量安排在它的背景数据块中，背景数据块不是由用户编辑的，而是由编辑器自动生成的。

用户也可以自己定义数据块（DB of Type），即以 UDT 为模板生成数据块。创建用户定义数据块之前，必须先创建一个用户定义数据类型，如 UDT1，并在 S7 程序编辑器内定义。

用户程序可通过位、字节、字或双字操作访问数据块中的数据，也可以使用符号或绝对地址访问。

2．数据块的建立

在 STEP 7 Manager 界面中，建立新项目，按照图 7-2 所示步骤建立数据块，步骤 3 操作

完毕后，弹出界面如图 7-3 所示，单击“确定”按钮，完成数据块的建立。

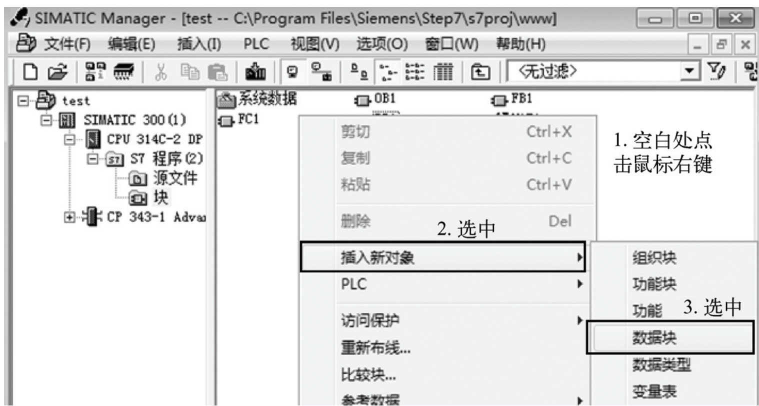


图 7-2　数据块的建立 1

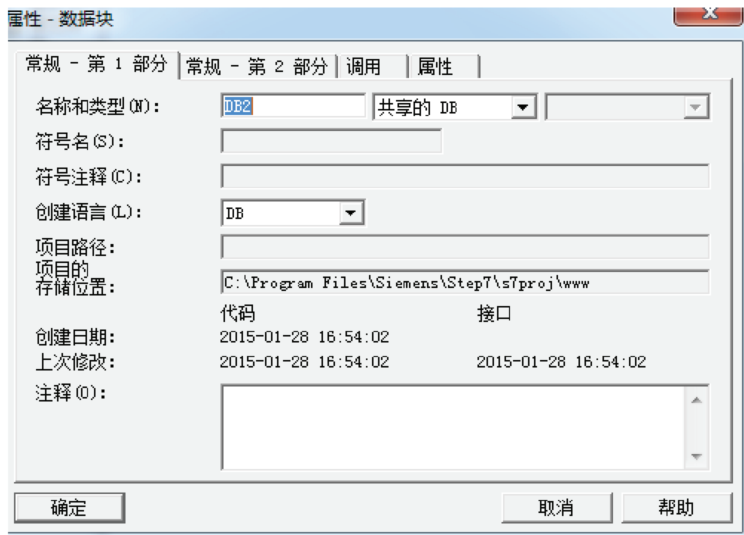


图 7-3　数据块的建立 2

3．数据块的数据结构

在 STEP 7 中数据块的数据类型可以采用基本数据类型、复杂数据类型或用户定义数据类型。

7.1.2　数据类型

1．基本数据类型

基本数据类型见表 7-1。

2．复杂数据类型

使用复杂数据类型是为了更好地构造数据，包括数组（ARRAY）、结构（STRUCT）、字符串（STRING）、日期和时间（DATE AND TIME）以及用户定义的数据类型（UDT）等。

表 7-1　基本数据类型

关　键　词	数据长度（位）	范围及表示方法	范　　例
BOOL（位）	1	0，1	1 或 0，True 或 False
BYTE（字节）	8	B#16#0 ~ B#16#FF	B#16#5B
WORD（字）	16	W#16#0 ~ W#16#FFFF BCD 码：C#0 ~ C#999	W#16#1234 C#235
DWORD（双字）	32	DW#16#0000_0000 ~ DW#16#FFFF_FFFF	DW#16#ABCD1234
INT（整数）	16	-32768 ~ +32767	123
DINT（双整数）	32	L#-214783648 ~ L#214783647	L#12345
REAL（浮点数）	32	±1.175495e-38 ~ ±3.402823e+38	5.6 或 1.52e–5
TIME（IEC 时间）	32	T#-24D_20H_31M_23S_648MS ~ T#24D_20H_31M_23S_647MS 分辨率为 1ms	T#2D_2H_1M_15S_25MS
DATE（IEC 日期）	16	D#1990_1_1 ~ D#2168_12_31 分辨率为 1day	D#2014-9-30
TIME_OF_DAY（时间）	32	TOD#0:0:0.0 ~ TOD#23:59:59:999 分辨率为 1ms	TOD#11:21:25.23
S5TIME（SIMATIC 时间）	16	S5T#0H_0M_0S_10MS ~ S5T#2H_46M_30S_0MS 分辨率 10ms	S5T#4S_10MS
CHAR	8	可打印的 ASCII 字符	‘A’；’B’；’0’

（1）数组（ARRAY）

数组是由一组同一类型的数据组合在一起而形成的复杂数据类型。数组的维数最大可以到 6 维；数组中的元素可以是基本数据类型或者复杂数据类型中的任一数据类型（Array 类型除外，即数组类型不可以嵌套）；数组中每一维的下标取值范围是-32768~32767，要求下标的下限必须小于下标的上限。

例如，建立一个数组，数组有 5 个元素且为整数，可以建立数据块 DB2，内容设置见表 7-2，每个整数占 2 个字节，该数组共占 10 个字节。

表 7-2　DB2 内容

地址	名称	类型	初始值	注释
0.0		STRUCT		
+0.0	test	array[1..5]	0	临时占位符变量
*2.0		INT		
=10.0		END_STRUCT		

图 7-4 所示是数组中的第 2 个字和第 4 个字的赋值程序，将其下载至 CPU 中。

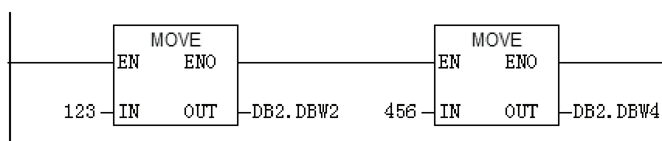


图 7-4　赋值程序

打开数据块 DB2，单击快捷键监视开关“”，弹出界面如图 7-5 所示，单击“是”按钮，弹出界面如图 7-6 所示。

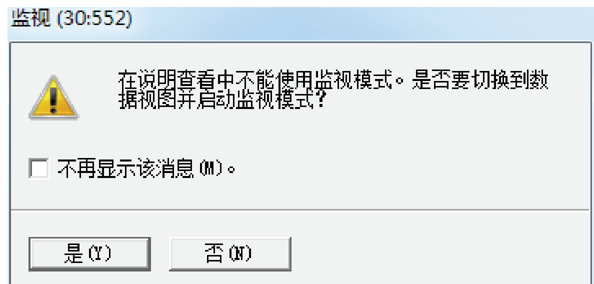


图 7-5　启动监视模式过程

由图 7-6 可见，数组中第 2 个变量和第 3 个变量状态已发生变化。同理可以修改或监控数组中的任何一个变量。

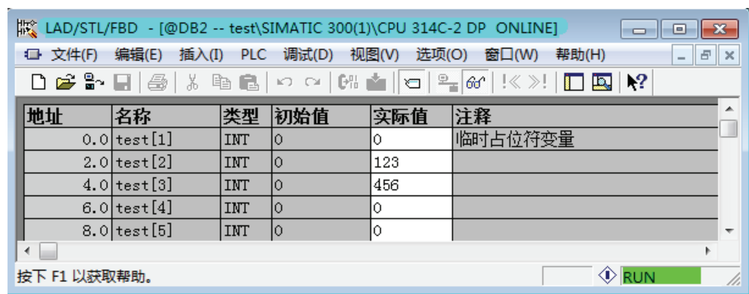


图 7-6　DB1 的变量监控

（2）结构（STRUCT）

结构是由一组不同类型（结构的元素可以是基本的或复杂的数据类型）的数据组合在一起而形成的复杂数据类型。结构通常用来定义一组相关的数据，例如被控电动机需要的参数包括启动、停止、设定速度及实际速度，建立数据表 DB3，按表 7-3 所示的结构定义一组数据。

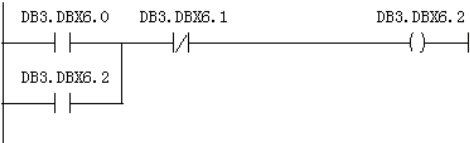
表 7-3　Motor 参数

DB3 -- test\SIMATIC 300(1)\CPU 314C-2 DP

地址	名称	类型	初始值
0.0		STRUCT	
+0.0	set_speed	INT	0
+2.0	actual_speed	REAL	0.000000e+000
+6.0	start	BOOL	FALSE
+6.1	stop	BOOL	FALSE
+6.2	km	BOOL	FALSE
=8.0		END_STRUCT	

调用结构中的变量程序如图 7-7 所示。当系统运行时，结构表中数据变化如图 7-8 所示。

程序段 4：起动、停止电动机



程序段 5：给定电动机速度

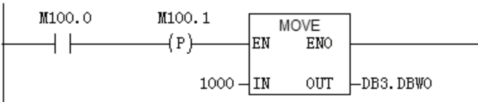


图 7-7　结构变量的调用

地址	名称	类型	初始值	实际值
0.0	set_speed	INT	0	1000
2.0	actual_speed	REAL	0.000000e+000	0.0
6.0	start	BOOL	FALSE	FALSE
6.1	stop	BOOL	FALSE	FALSE
6.2	km	BOOL	FALSE	TRUE

图 7-8　结构变量监视

（3）字符串（STRING）

字符串用来处理消息文本，是最多有 254 个字符（CHAR）的一维数组，最大长度为 256 个字节（其中前两个字节用来存储字符串的长度信息）。字符串常量用单引号括起来，例如，实现‘HELLO’的显示。建立数据表 DB4，内容如图 7-9 所示，编写一段变量传递程序以便观察变量数值，如图 7-10 所示。

地址	名称	类型	初始值
0.0		STRUCT	
+0.0	H	CHAR	'H'
+1.0	E	CHAR	'E'
+2.0	L	CHAR	'L'
+3.0	L1	CHAR	'L'
+4.0	O	CHAR	'O'
=6.0		END_STRUCT	

图 7-9　字符串数据块

程序段 6：字符串的应用

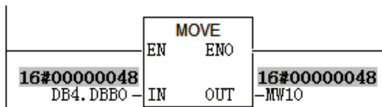


图 7-10　读出第一个字符的编码

从图 7-10 可见，读出的第一个字符编码为 16#48，通过图 7-11 所示的变量表监控数据可见，字符‘H’的十六进制编码是 48。

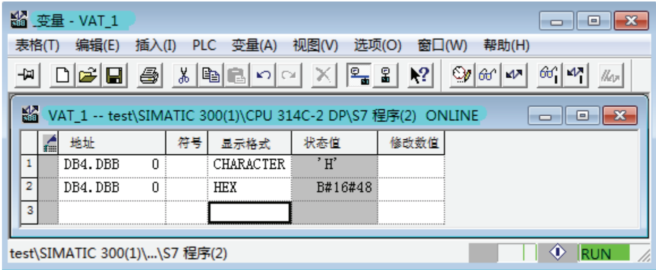


图 7-11　变量表监控变量

（4）日期和时间（DATE_AND_TIME）

该类数据类型用于存储年、月、日、时、分、秒、毫秒和星期，占用 8 个字节，用 BCD 格式保存。星期天的代码为 1，星期一~星期六的代码为 2~7。例如建立数据块 DB5 并打开数据块，在数据块中建立一个 DT 变量，初始值自动生成，如图 7-12 所示。

地址	名称	类型	初始值	注释
0.0		STRUCT		
+0.0	day_test	DATE_AND_TIME	DT#90-1-1-0:0:0.000	临时占位符变量
=8.0		END_STRUCT		

剪切(T) Ctrl+X
复制(C) Ctrl+C
粘贴(P) Ctrl+V
删除(L) Del
基本类型(E)
复杂类型(X)
对象属性(O) Alt+Return

DATE_AND_TIME
STRING
ARRAY
STRUCT
UDT

图 7-12　DT 变量的建立

如图 7-13 所示，DB5 中共建立两个 DT 变量，每个变量占用 8 个字节，共 16 个字节。

LAD/STL/FBD - [DB5 -- test\SIMATIC 300(1)\CPU 314C-2 DP]

文件(F) 编辑(E) 插入(I) PLC 调试(D) 视图(V) 选项(O) 窗口(W) 帮

地址	名称	类型	初始值
0.0		STRUCT	
+0.0	day_test1	DATE_AND_TIME	DT#90-1-1-0:0:0.000
+8.0	day_test2	DATE_AND_TIME	DT#90-1-1-0:0:0.000
=16.0		END_STRUCT	

图 7-13　DB5 数据块内容的建立

如图 7-14 所示，在 OB1 中编写系统时间设定和读出程序。图 7-15 所示为在线修改系统时间和读出系统时间的变量监控表。

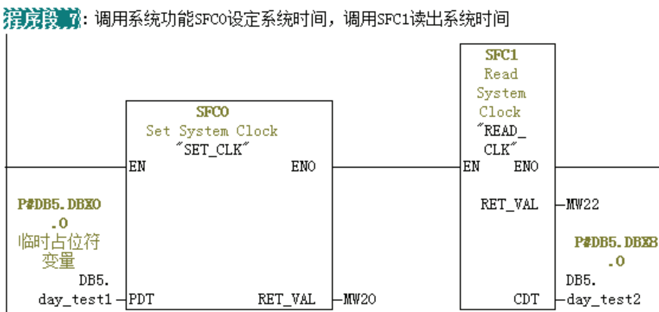


图 7-14　程序编写

变量 - [VAT_1 -- @test\SIMATIC 300(1)\CPU 314C-2 DP\S7 程序(2) ONLI...

表格(T) 编辑(E) 插入(I) PLC 变量(A) 视图(V) 选项(O) 窗口(W) 帮助(H)

	地址		符号	显示格式	状态值	修改数值
1	DB5.DBW	0		HEX	W#16#1501	W#16#1501
2	DB5.DBW	2		HEX	W#16#2411	W#16#2411
3	DB5.DBW	4		HEX	W#16#3200	W#16#3200
4	DB5.DBW	6		HEX	W#16#0000	W#16#0000
	DB5.DBW	8		HEX	W#16#1501	
6	DB5.DBW	10		HEX	W#16#2411	
7	DB5.DBW	12		HEX	W#16#3200	
8	DB5.DBW	14		HEX	W#16#0007	

设定当前日期

读出系统时间

test\SIMATIC 300(1)\...\S7 程序(2)　RUN

图 7-15　系统运行数据

（5）用户定义的数据类型（UDT）

用户自定义数据类型（由基本数据类型及复杂数据类型组成的模板）存放在 UDT 块中（UDT1~UDT65535），在另一个数据类型中作为一个数据类型“模板”。例如，每台机床需要监控的内容相同，则在 STEP 7 管理器界面的程序结构中选择“插入新对象”→“数据类型”，建立 UDT1 数据块并在其中建立相应的变量，见表 7-4。

表 7-4　UDT1 变量内容

地址	名称	类型	初始值	注释
0.0		STRUCT		
+0.0	Beat_MT	BOOL	FALSE	网络信号
+0.1	Rdy_MT	BOOL	FALSE	准备
+0.2	InProgress_MT	BOOL	FALSE	加工中
+0.3	Waiting_MT	BOOL	FALSE	待料中
=2.0		END_STRUCT		

其他几台机床只需在全局符号表中为 UDT1 分配一个符号名就可以了，如图 7-16 所示建立一个数据块 DB1，在其中建立了两个相同结构变量 Auto_line_MT 及 Hand_line_MT 的数组，两个区域共有 8 台机床。

DB1 -- test\SIMATIC 300(1)\CPU 314C-2 DP

地址	名称	类型	初始值
0.0		STRUCT	
+0.0	Auto_line_MT	ARRAY[1..5]	
*2.0		UDT1	
+10.0	Hand_line_MT	ARRAY[1..3]	
*2.0		UDT1	
=16.0		END_STRUCT	

图 7-16　UDT1 的应用

图 7-17 所示为 DB1 数据的在线监控状态。可见，当输入数据块时，如果需要输入几个相同的结构，利用 UDT 既可以节省输入时间又方便变量的管理。

LAD/STL/FBD - [@DB1 -- test\SIMATIC 300(1)\CPU 314C-2 DP ONLINE]

文件(F)　编辑(E)　插入(I)　PLC　调试(D)　视图(V)　选项(O)　窗口(W)　帮助(H)

地址	名称	类型	初始值	实际值
0.0	Auto_line_MT[1].Beat_MT	BOOL	FALSE	FALSE
0.1	Auto_line_MT[1].Rdy_MT	BOOL	FALSE	TRUE
0.2	Auto_line_MT[1].InProgress_MT	BOOL	FALSE	FALSE
0.3	Auto_line_MT[1].Waiting_MT	BOOL	FALSE	FALSE
2.0	Auto_line_MT[2].Beat_MT	BOOL	FALSE	FALSE
2.1	Auto_line_MT[2].Rdy_MT	BOOL	FALSE	FALSE
2.2	Auto_line_MT[2].InProgress_MT	BOOL	FALSE	FALSE
2.3	Auto_line_MT[2].Waiting_MT	BOOL	FALSE	FALSE

图 7-17　DB1 块变量在线监控

3．参数数据类型

参数类型是一种用于逻辑块（FB、FC）之间传递参数的数据类型，如图 7-18 所示。

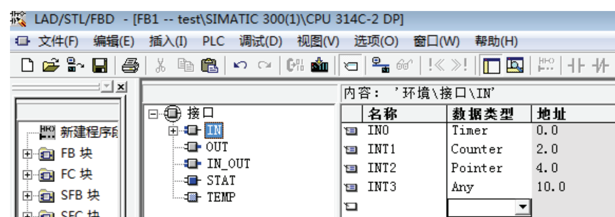


图 7-18　参数变量的建立

在建立 FB、FC 接口参数时，主要有以下几种类型：

1）TIMER（定时器）和 COUNTER（计数器）。

2）BLOCK（块）：指定一个块用作输入和输出，实参应为同类型的块。

3）POINTER（指针）：6 字节指针类型，用来传递 DB 的块号和数据地址。

4）ANY：10 字节指针类型，用来传递 DB 块号、数据地址、数据数量以及数据类型。

7.1.3　地址区域

S7-300 CPU 的系统存储器划分的地址区域见表 7-5，在用户程序中使用指令可以在相应的地址区直接对数据进行寻址。

表 7-5　地址区域及功能

存储器	单位	标识符	寻址范围	功能描述
过程映像输入	输入位	I	0.0~65535.7	在每次循环周期扫描（OB1）开始时，CPU 读取输入模块的输入信号并将数值存入该区域。例如：I0.0，IB10，IW20，ID16
	输入字节	IB	0~65535	
	输入字	IW	0~65534	
	输入双字	ID	0~65532	
过程映像输出	输出位	Q	0.0~65525.7	在循环周期扫描过程中，将程序计算的输出值存放在该区域，在扫描结束时，CPU 将这些输出值发送到输出模块。例如：Q0.0，QB5，QW8，QD30
	输出字节	QB	0~65535	
	输出字	QW	0~65534	
	输出双字	QD	0~65532	
位存储器	存储位	M	0.0~255.7	用于存放程序运行的中间结果。例如：M0.0，MB3，MW6，MD10
	存储字节	MB	0~255	
	存储字	MW	0~254	
	存储双字	MD	0~252	
定时器	定时器（T）	T	0~255	定时器存储区域。例如：T0，T1，…
计数器	计数器（C）	C	0~255	计数器存储区域。例如：C0，C1，…
共享数据块 DB	数据位	DBX	0.0~65525.7	共享数据块中包含程序需要的信息，可以为所有逻辑块通用，使用 OPN DB 指令打开一个共享数据块。例如：DB0，DBX10.0，DBW20，DBD100
	数据字节	DBB	0~65535	
	数据字	DBW	0~65534	
	数据双字	DBD	0~65532	
背景数据块 DI	数据位	DIX	0.0~65525.7	背景数据块中包含程序需要的信息，由某个特定的 FB 或 SFB 使用，使用 OPN DI 指令打开一个背景数据块。例如：DIX0.0，DIB10，DIW20，DID50
	数据字节	DIB	0~65535	
	数据字	DIW	0~65534	
	数据双字	DID	0~65532	

（续）

存　储　器	单　　位	标　识　符	寻 址 范 围	功 能 描 述
局域数据	局域数据位	L	0.0~65525.7	在处理组织块、功能块和系统功能块时，相应块的临时数据保存到该块的局部数据区域。 例如：L0.2，LB2，LW10，LD30
	局域数据字节	LB	0~65535	
	局域数据字	LW	0~65534	
	局域数据双字	LD	0~65532	
外设输入区	外设输入字节	PIB	0~65535	通过该区域，用户可立即直接访问输入模块。 例如：PIB0，PIW200，PID300
	外设输入字	PIW	0~65534	
	外设输入双字	PID	0~65532	
外设输出区	外设输出字节	PQB	0~65535	通过该区域，用户可立即直接访问输出模块。 例如：PQW256，PQW280，PQW320
	外设输出字	PQW	0~65534	
	外设输出双字	PQD	0~65532	

从表中可见，I、Q、M、L 有位寻址、字节寻址、字寻址及双字寻址几种方式，PI 和 PQ 只有字节寻址、字寻址及双字寻址，没有位寻址方式。

7.1.4　指令操作数

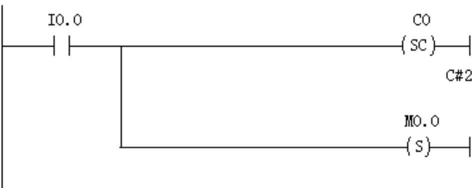
指令操作数（又称编程元件）一般在用户存储区中，操作数由操作标识符和参数组成。操作标识符由主标识符和辅助标识符组成，主标识符用来指定操作数所使用的存储区类型，辅助标识符则用来指定操作数的单位（如：位、字节、字、双字等）。

主标识符有：I、Q、M、PI、PQ、T、C、DB 和 L。

辅助标识符有：X、B、W、D。

用法如图 7-19 所示。

程序段 8：标题：



程序段 9：标题：



程序段 10：标题：

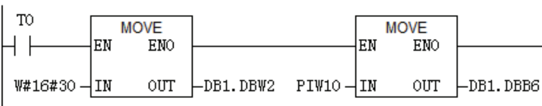


图 7-19　操作数的用法

7.1.5 CPU 中的寄存器

1. 累加器

32 位累加器是用来处理字节、字和双字的寄存器。S7-300 PLC 有两个累加器，分别是 ACCU1 和 ACCU2，可以把操作数装入累加器并进行运算和处理，保存在 ACCU1 中的运算结果可以传送到系统存储器中。

2. 16 位状态寄存器

状态字是一个 16 位的寄存器，用于存储 CPU 在执行指令的过程中产生的状态。一些指令是否执行或以何方式执行可能取决于状态字中的某些位，执行指令时也可能改变状态字中某些位的状态，也可以在位逻辑指令或字逻辑指令中访问并检测状态字，状态字的结构见表 7-6。

表 7-6 状态字结构

位　序	符　号	名　称	含　义
15~9	—		—
8	BR	二进制结果位	1）用来表示字操作结果是否正确。在 LAD 的方块指令中，BR 位与 ENO 有对应关系，如果方块执行错误，BR 位为 0，ENO 也等于 0；如果功能被正确执行，BR=1，ENO 也等于 1。 2）在用户用语句表编写 FB 和 FC 程序中，必须对 BR 位进行管理；当 FB 或 FC 执行无错误时，RLO 为 1，并可以用 SAVE 指令将 RLO 存入 BR，保证能流导通
7	CC1	条件码 1	这两位结合起来用于表示在累加器 1 中产生的算术运算或逻辑运算结果与“0”的大小关系
6	CC0	条件码 0	
5	OS	溢出状态保持位	OV 被置“1”时，OS 也被置“1”，OV 被清 0 时，OS 仍保持，只有 JOS、块调用和块结束指令可以复位 OS 位
4	OV	溢出位	为“1”时表明一个算术运算或浮点数比较指令执行时出现错误，例如溢出、非法操作、不规范格式
3	OR	或位	在先逻辑“与”后逻辑“或”的逻辑串中，OR 位暂存逻辑“与”结果，以便进行后面的逻辑“或”运算，其他指令将 OR 位清“0”
2	STA	状态位	该位不能用指令检测，它只是在程序测试中被 CPU 解释并使用
1	RLO	逻辑操作结果	该位存储位逻辑指令或算术比较指令的结果
0	/FC	首次检测位（斜杠表示对 FC 取反）	若为“0”，则表明一个梯形逻辑网络的开始，或指令为逻辑串第一条指令

7.1.6 寻址方式

所谓寻址方式就是指令执行时获取操作数的方式，可以直接或间接方式给出操作数。S7-300 有 4 种寻址方式。

1．立即寻址

立即寻址是对常数或常量的寻址方式，其特点是操作数直接表示在指令中，或以唯一形式隐含在指令中。如图 7-20 所示的指令操作数中，INT1、INT2 及 IN 均采用了立即寻址方式。

2．存储器直接寻址

存储器直接寻址，简称直接寻址。该寻址方式在指令中直接给出操作数的存储单元地址，如图 7-21 所示，存储单元地址可用绝对地址（如 I0.1、Q0.0）或符号地址（如“速度设定”、“电动机控制.set_speed”等）。图 7-21 中的各条指令操作数均采用了直接寻址方式。



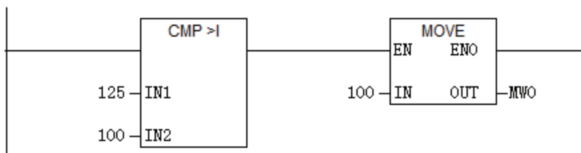


图 7-20　立即寻址

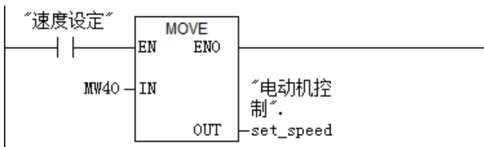


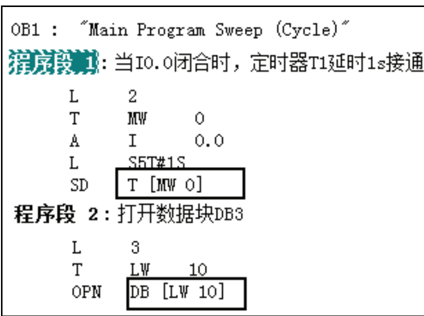
图 7-21　存储器直接寻址

3．存储器间接寻址

存储器间接寻址，简称间接寻址。该寻址方式在指令中以存储器的形式给出操作数所在存储器单元的地址，也就是说该存储器的内容是操作数所在存储器单元的地址。该存储器一般称为地址指针，在指令中需写在方括号“[ ]”内。

存储器地址指针分为 16 位地址指针和 32 位地址指针。

图 7-22 所示为 16 位指针存储器间接寻址方式。程序段 1 指令的含义是，将常数 2 送入 MW0 中，MW0 的值用来指定定时器的地址，即定时器地址为 T2，T 为区域表示符，而[MW0]为一个 16 位指针；程序段 2 指令的含义是，将常数 3 送到地址 LW10 中，LW10 的值用来指定要打开的数据块地址 DB3，DB 为区域表示符，而[LW10]为一个 16 位指针。



```
OB1 :  "Main Program Sweep (Cycle)"
程序段 1：当I0.0闭合时，定时器T1延时1s接通
    L     2
    T     MW     0
    A     I      0.0
    L     S5T#1S
    SD    T  [MW 0]
程序段 2：打开数据块DB3
    L     3
    T     LW     10
    OPN   DB  [LW 10]
```

图 7-22　16 位指针存储器间接寻址

16 位地址指针用于定时器、计数器和程序块（DB/FC/FB）的寻址，被看作为一个无符号整数（0~65535），它表示定时器、计数器、数据块（DB/DI）或程序块（FB/FC）的地址号。

32 位地址指针用于 I、Q、M、L 以及数据块等存储器中位、字节、字及双字的寻址，32 位的地址指针可以使用一个双字表示，第 0~2 位作为寻址操作的位地址，第 3~18 位作为寻址操作的字节地址，第 19~31 位没有定义。

32 位地址指针仅用于内部区域寻址。寻址格式如下：

地址存储区标示符[32 位地址指针]

例如，编写程序指令如下：

```
程序段 1：标题：
L    P#2.3          //将指针值载入ACCU1
T    MD     0       //将指针传送到MD0中
A    I  [MD 0]      //扫描输入位I2.3
=    Q  [MD 0]      //将I2.3状态分配给输出位Q2.3
```

其中“P#2.3”为常数指针，意为指向第 2 个字节第 3 位。可在线观察地址指针内容，如图 7-23 所示。

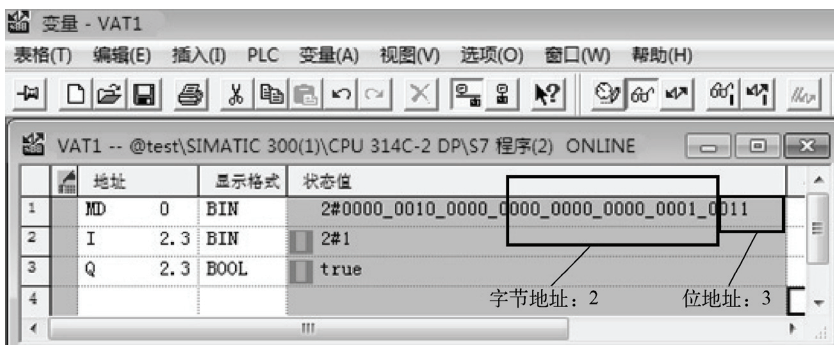


图 7-23　在线观察指针内容

4．寄存器间接寻址

与存储器间接寻址不同，寄存器间接寻址使用 CPU 集成的两个 32 位地址寄存器 AR1 和 AR2 存储地址指针，且只能用于存储地址。

寄存器间接寻址，简称寄存器寻址。该寻址方式在指令中通过地址寄存器和偏移量间接获取操作数，其中的地址寄存器及偏移量必须写在方括号“[ ]”内，用地址寄存器的内容加上偏移量形成地址指针，并指向操作数所在的存储器单元。地址寄存器的地址指针有两种格式，其长度均为双字。

（1）区域内寄存器间接寻址

这种地址指针格式适用于在确定的存储区内寻址，即区内寄存器间接寻址，指针格式与存储器间接寻址的 32 位指针相同，即：第 0~2 位作为寻址操作的位地址，第 3~18 位作为寻址操作的字节地址，第 19~31 位没有定义，操作指令格式如下：

存储器标示符[ARx，地址偏移量]

图 7-24 所示为一段区域内寄存器间接寻址程序，AI[AR1，P#0.1]。‘I’为被访问的存储器区域，‘AR1’为地址寄存器 1，P#0.1 为偏移地址。

程序段 3：标题：

```
L      P#2.3           //将指针值装入ACCU1中;
LAR1                   //用ACCU1中的指针装载AR1;
A      I [AR1,P#0.1]   //AR1地址2.3，偏移量不会对其产生影响，但扫描输入位地址为AR1+偏移量，即I2.4;
=      Q [AR1,P#0.2]   //将I2.4状态位信息分配给输出位Q2.5
```

图 7-24　区域内寄存器指针应用案例

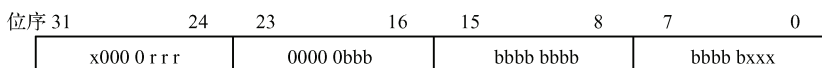
区域内寄存器间接寻址的特点是：地址标示符在方括号前确定，方括号中的指针不含存储区信息，否则会与方括号前的地址标示符标示的存储区冲突。

区域内寄存器间接寻址包括被寻址数据所在存储单元地址的字节编号和位编号，至于对哪个存储区寻址，则必须在指令中明确给出。这种格式适用于在确定的存储区内寻址。

（2）区域间寄存器间接寻址

这种地址指针格式适用于跨区域寻址，包含有存储器区域信息的指针，称为交叉区域指针，要使用 AR1 或 AR2。

交叉区域指针为 32 位，左起 0~18 位格式与 32 位区域内寄存器指针格式相同，19~23 位、27~30 位未定义，31 位为交叉区域指针标示符，具体格式及含义如图 7-25 所示。



说明：位0～2(xxx)为被寻址地址中位的编号(0～7)
位3～18为被寻址地址的字节的编号(0～65535)
位24～26(rrr)为被寻址地址的区域标识号
位31的x=0为区域内的间接寻址，x=1为区域间的间接寻址

图 7-25　交叉区域指针格式

其中第 24~26 位为存储区域地址标识，8 种组合表示 8 种存储器区域，见表 7-7。

表 7-7　地址区域及功能

Bit（26/25/24）	000	001	010	011	100	101	110	111
区域标识	常数	I	Q	M	DB	DI	L	V

图 7-26 所示为一段交叉区域寄存器间接寻址程序，其中“= [AR2，P#0.2]”，‘AR2’为地址寄存器 2，P#0.2 为偏移量。图 7-27 所示为该段程序的在线监控表，从表中可看到地址寄存器 AR1/AR2 的数据格式。

程序段 2：标题：

```
L     P#I 0.1          //将指针值装入ACCU1中；
LAR1                   //将I0.1存入AR1中；
T     MD    10         //MD10读出AR1的地址值以便观察格式；
L     P#Q 3.1          //将指针值装入ACCU1中；
LAR2                   //将Q3.1存入AR2中；
T     MD    14         //MD14读出AR2的地址值以便观察数据格式；
A      [AR1,P#0.0]     //扫描输I0.1；
=      [AR2,P#0.2]     //将I0.1状态位信息分配给输出位Q3.3。
```

图 7-26　跨区域寄存器指针应用案例

变量 - [VAT1 -- @test\SIMATIC 300(1)\CPU 314C-2 DP\S7 程序(2) ONLINE]

表格(T)　编辑(E)　插入(I)　PLC　变量(A)　视图(V)　选项(O)　窗口(W)　帮助(H)

	地址	显示格式	状态值	修改数值
1	I 0.1	BIN	2#1	
2	Q 3.3	BOOL	true	
3	MD 10	BIN	2#1000_0001_0000_0000_0000_0000_0000_0001	
4	MD 14	BIN	2#1000_0010_0000_0000_0000_0000_0001_1001	
5				

标识符：I　第0个字节　位编号　标识符：Q　第3个字节

图 7-27　变量在线监控

跨区域寄存器间接寻址的特点是：地址标示符在方括号前确定，方括号中的指针包含存储区信息。由于跨区域寄存器间接寻址在指令中不确定存储区，存储区由地址寄存器中存储的指针中的信息确定，因而寻址的存储区是可变的。

跨区域寄存器间接寻址指针格式包含了数据所在存储区的说明位（存储区域标识位），可通过改变标识位实现跨区域寻址，区域标识由第 26~24 位确定。

7.2 位逻辑指令

位逻辑指令处理的对象为二进制信号。对于触点和线圈而言，“0”表示未激活或未励磁，“1”表示已激活或已励磁。

位逻辑指令解释信号状态“0”和“1”，并根据布尔逻辑对它们进行组合，所产生的结果称为逻辑运算结果，存储在状态字的“RLO”中。

7.2.1 触点指令

触点指令包括基本指令及执行嵌套表达式时的嵌套指令，指令符号及描述见表 7-8。

表 7-8 触点指令及含义

基本指令	含义	嵌套指令	含义
A	与运算	A（	与运算嵌套开始
AN	与非运算	AN（	与非运算嵌套开始
O	或运算	O（	或运算嵌套开始
ON	或非运算	ON（	或非运算嵌套开始
X	异或运算	X（	异或运算嵌套开始
XN	同或运算	XN（	同或运算嵌套开始
		）	嵌套结束

【例 7-1】 基本指令应用 1。如图 7-28 所示。

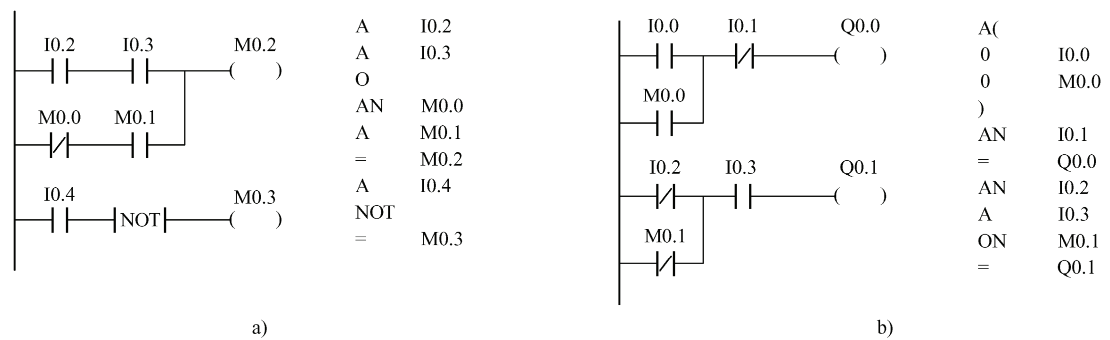


图 7-28 基本指令应用 1

a) 基本指令连接示例 b) 嵌套指令使用示例

说明：在示例中，“=”为赋值指令，将逻辑运算结果写入地址位，功能与 S7-200 的“=”相当，使用的操作数为 Q、M、L、D；“｜NOT｜”为逻辑取反指令，用来将其左边电路的逻辑运算结果 RLO 取反，与 S7-200 的“｜NOT｜”指令相当。

【例 7-2】 基本指令应用 2。如图 7-29 所示。

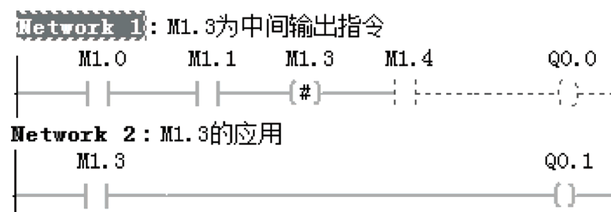


图 7-29　基本指令应用 2

说明：在示例中，“(#)”为中间输出单元指令，该单元用于保存分支单元的逻辑结果，以串联方式与其他触点连接时，可以像插入触点那样插入 ---(＃)--，M1.3 的逻辑等于（M1.0 AND M1.1）。

【例 7-3】 基本指令应用 3。如图 7-30 所示。

说明：在示例中，“(SAVE)”指令将逻辑运算结果 RLO 的状态保存到状态字的 BR 位，BR 位的状态包含在下一程序段的 AND 逻辑运算中。如图 7-30a 所示，当 M2.0=1 时，RLO=1，BR=1，则下一程序段的逻辑运算为(BR) AND (M2.1)=1，所以 Q0.2=1；当 M2.0=0 时，RLO=0，BR=0，则下一程序段的逻辑运算为(BR) AND (M2.1)=0。

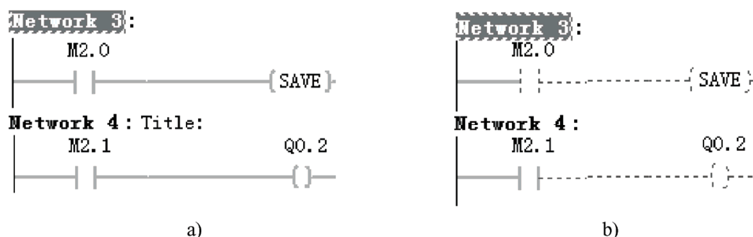


图 7-30　基本指令应用 3

a) M2.0 接通　b) M2.0 断开

7.2.2　置位和复位指令

置位指令（S）的功能是将位存储区置 1；复位指令（R）的功能是将位存储区置 0。如图 7-31 所示。

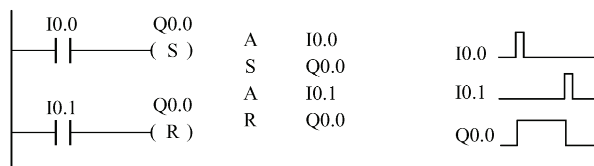


图 7-31　S/R 指令使用示例

7.2.3　RS 和 SR 触发器

RS 触发器工作梯形图如图 7-32a 所示。当 I0.0=1、I0.1=0 时，M0.0=0，Q0.0=0；当 I0.0=0、I0.1=1 时，M0.0=1，Q0.0=1；当 I0.0=1、I0.1=1 时，由于程序周期扫描顺序是先扫描复位后扫描置位，因此 M0.0=1，Q0.0=1，即置位优先。

SR 触发器工作梯形图如图 7-32b 所示。当 I0.2=1、I0.3=0 时，M0.1=1，Q0.1=1；当 I0.2=0、I0.3=1 时，M0.1=0，Q0.1=0；当 I0.2=1、I0.3=1 时，由于程序周期扫描顺序是先扫描置位后扫描复位，因此 M0.1=0，Q0.1=0，即复位优先。

RS 触发器和 SR 触发器的 S 端、R 端及 Q 端所使用的操作数可以是 I、Q、M、L、D。

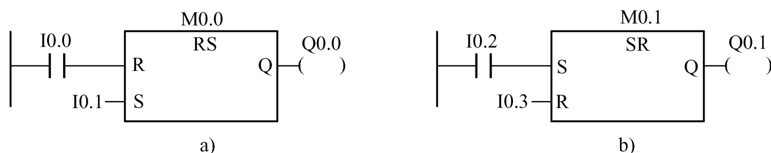


图 7-32　触发器使用示例

a) RS 触发器　b) SR 触发器

7.2.4　跳变沿检测指令

STEP 7 中有 2 类跳变沿检测指令，一类是对 RLO 的跳变沿检测的指令，另一类是对触点的跳变沿直接检测的梯形图方块指令。

【例 7-4】 对 RLO 的跳变沿检测示例。

这种跳变沿检测指令用来检测逻辑运算结果 RLO 的跳变。在图 7-33a 中，M0.0 用来储存其左边的 RLO 逻辑运算结果（I0.0），中间标有“P”的上升沿检测 RLO 是否有从“0”到“1”的正跳变，如果有，则 Q0.0 导通一个扫描周期；在图 7-33b 中，M0.1 用来存储其左边的 RLO 逻辑运算结果（I0.2*I0.3），中间标有“N”的下降沿检测 RLO 是否有从“1”到“0”的负跳变，如果有，则 Q0.1 导通一个扫描周期。

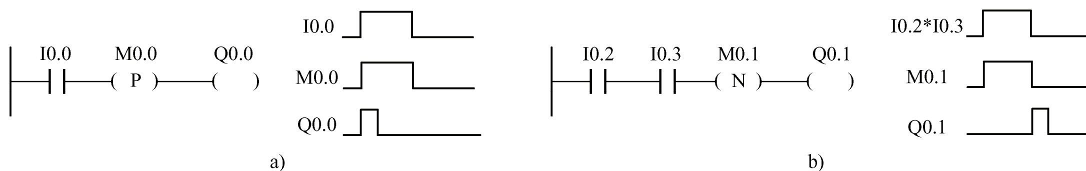


图 7-33　对 RLO 的跳变沿检测指令示例

a) 上升沿　b) 下降沿

【例 7-5】 触点信号跳变沿检测指令示例。

这种跳变沿检测指令用来检测单个地址位的跳变。在图 7-34a 中，I0.0 为上升沿指令导通条件，M0.1 为边沿存储位，存储上一次循环时 M0.0 的状态，当 M0.0 由“0”到“1”跳变时，Q0.0 接通 1 个扫描周期；在图 7-34b 中，I0.1 为下降沿指令导通条件，M0.3 为边沿存储位，存储上一次循环时 M0.2 的状态，当 M0.2 由“1”到“0”跳变时，Q0.1 接通 1 个扫描周期。

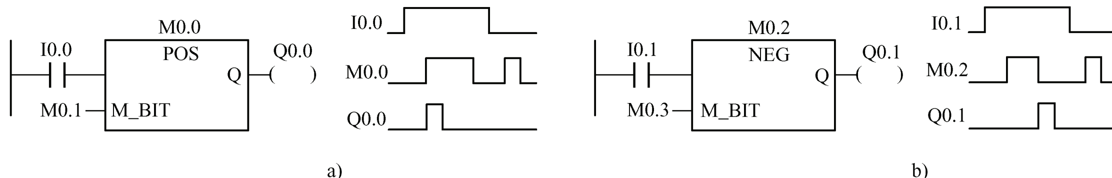


图 7-34　对触点的跳变沿检测指令示例

a) 上升沿　b) 下降沿

7.2.5　SET/CLR 指令

SET/CLR 指令可以更改逻辑运算结果 RLO，将 RLO 置位或复位，紧接其后的赋值语句中的地址将变为“1”状态或“0”状态，使用格式如程序段 4：

```
程序段 4：标题：
    SET                                 //将RLO置位
    =     M      0.0                    //M0.0状态为“1”
    CLR                                 //将RLO复位
    =     M      0.1                    //M0.1状态为“0”
```

7.3　定时器指令

存储器中的定时器区域用来存储定时器的定时时间值。每一个定时器占 2 个字节，称为定时字。在 S7-300 系列 PLC 中，定时区为 512 个字节，故只能使用 256 个定时器。定时器的访问有相关的定时器指令，其编址方式为 T 加编号，如 T22、T200 等。S7-300 PLC 分为五类定时器，分别是脉冲定时器（SP）、扩展脉冲定时器（SE）、接通延时定时器（SD）、保持型接通延时定时器（SS）和断开延时定时器（SF）。也可以采用 S5 定时器，S5 系列 PLC 的定时器，在梯形图中用指令框（BOX）的形式来表示。

7.3.1　脉冲定时器（SP）

脉冲定时器用于产生指定时间脉冲宽度。如图 7-35a 所示，T0 表示要启动的定时器号或定时器地址，S5T#5S 是 S5 TIME 格式（S5T#H_M_S_MS），表明定时器从启动开始运行 5s，当 5s 时间到，动合触点 T0 断开，Q0.0 失电；如果运行期间定时器导通条件 I0.0 断开，则定时器停止工作，Q0.0 维持的脉冲宽度也小于 5s，即脉冲定时器输出的高电平宽度小于等于所定义的时间值；I0.1 为定时器复位信号，如果 I0.1 有一个从“0”到“1”的变化，即使导通条件存在，定时器也会复位。

图 7-35b 为 S5 脉冲定时器，其功能、输入输出波形与图 7-35a 一样，可以从变量 MW0、MW2 中读出定时器 T0 定时剩余的时间；S 端为指令框导通条件，TV 端为时间设定值，BI 端为输出不带时间基准的十六进制格式当前时间值，BCD 端为输出 S5T#格式的当前时间值。

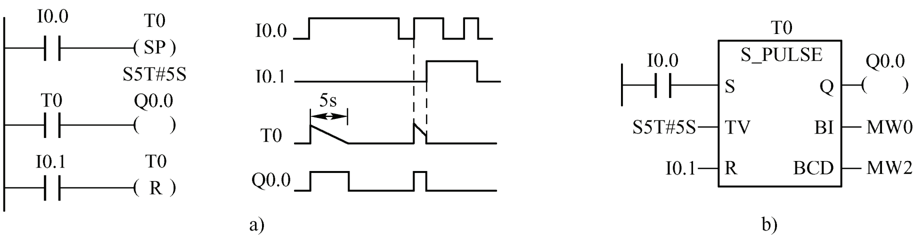


图 7-35　脉冲定时器工作示例

【例 7-6】 当条件满足时（M0.0），Q0.0 输出占空比为 25%的脉冲信号。程序设计如图 7-36 所示。

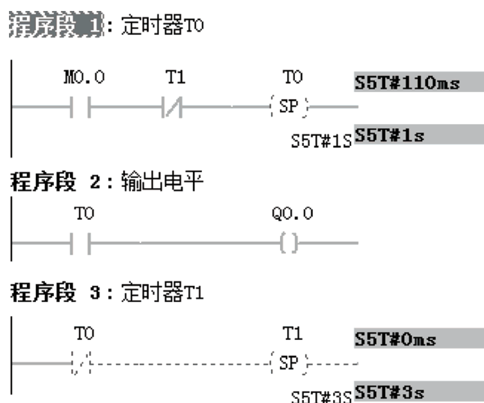


图 7-36　脉冲定时器应用示例

7.3.2　扩展脉冲定时器（SE）

扩展脉冲定时器指令也用于产生指定时间脉冲宽度。但与脉冲定时器不同的是，扩展脉冲定时器被触发后，输出值在定时时间内始终为 1，过了定时时间后，输出值为 0；即该类定时器与触发条件导通的时间长短无关，如图 7-37a 所示。

图 7-37b 所示为 S5 扩展脉冲定时器，其功能和输入输出波形与图 7-37a 一样，各输入/输出端子含义与 S5 脉冲定时器相同。

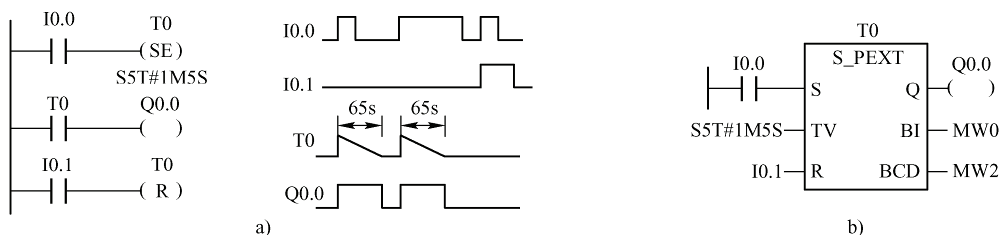


图 7-37　扩展脉冲定时器工作示例

【例 7-7】 当按下起动按钮 SB1（I0.0），电动机 M（Q0.0）立即起动并连续运转，延时 2min 后电动机停止；当按下停止按钮 SB2（I0.1），电动机 M 立即停止。程序实现如图 7-38 所示。

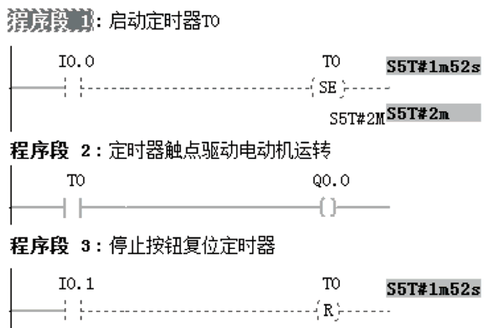


图 7-38　扩展脉冲定时器指令使用示例

7.3.3　接通延时定时器（SD）

接通延时定时器（SD）是使用最广的定时器，用于定时器触点延时接通驱动负载的电路。如图 7-39a 所示，定时器在导通条件 I0.0 为“1”时开始运行，在运行过程中定时器计时时间达到设定值 5s 时，输出 Q0.0 为“1”并保持，直到导通条件消失（I0.0=0）或按下复位按钮（I0.1=1）定时器复位；在计时期间如果导通条件消失或复位按钮按下，则定时器停止计时或复位，当导通信号 I0.0=1 时，定时器重新开始计时。

图 7-39b 为 S5 接通延时定时器，其功能、输入输出波形与图 7-39a 一样，各输入/输出端子含义与 S5 脉冲定时器相同。

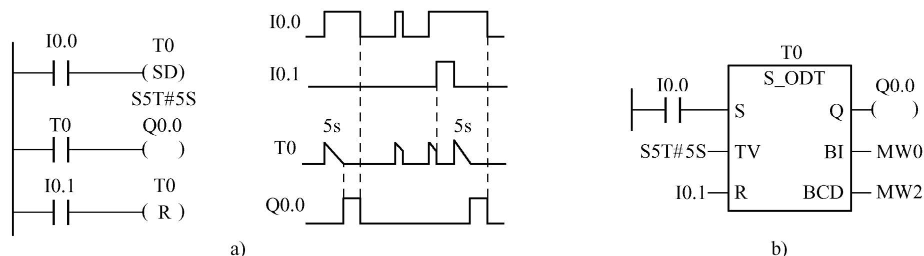


图 7-39　接通延时定时器工作示例

7.3.4　保持型接通延时定时器（SS）

与接通延时定时器相比，保持型接通延时定时器（SS）即使接通条件是短脉冲，定时器触点也能按照设定的时间导通。如图 7-40a 所示，定时器在导通条件 I0.0 由“0”变为“1”时开始运行（与接通时间长短无关），在运行过程中定时器计时时间到设定值 5s 时，输出 Q0.0 为“1”并保持，直到按下复位按钮（I0.1=1），Q0.0=0；在计时过程中，当导通条件重新由“0”变为“1”时，定时器重新开始计时；在计时期间或 Q0.0=1 期间，如果复位按钮按下，则定时器复位且 Q0.0=0。

图 7-40b 为 S5 保持型接通延时定时器，其功能、输入输出波形与图 7-40a 一样，各输入/输出端子含义与 S5 脉冲定时器相同。

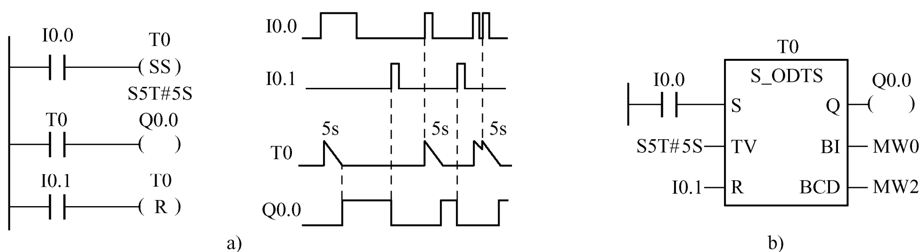


图 7-40　保持型接通延时定时器工作示例

7.3.5　断开延时定时器（SF）

断开延时定时器（SF）用于当导通条件有从“1”到“0”的负跳变时，定时器启动。如

图 7-41a 所示，定时器在导通条件 I0.0=1 时，Q0.0=1；当 I0.0 由“1”变为“0”时开始计时，在此期间 Q0.0=1，当定时器计时时间到达设定值 5s 时，输出 Q0.0=0；当按下复位按钮（I0.1=1），无论定时器是在计时期间还是导通期间，定时器复位，Q0.0=0；在计时过程中，当导通条件重新由“1”变为“0”时定时器重新开始计时。

图 7-41b 为 S5 断开延时定时器，其功能、输入输出波形与图 7-41a 一样，各输入/输出端子含义与 S5 脉冲定时器相同。

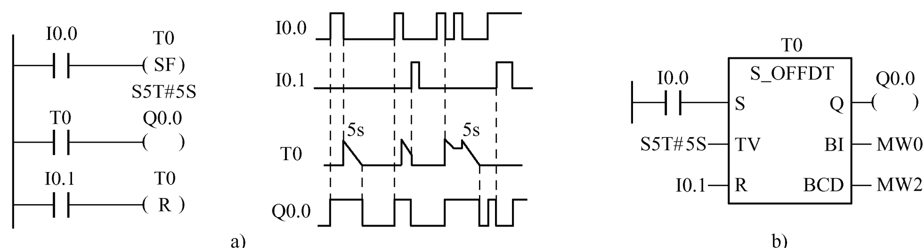


图 7-41　断开延时定时器工作示例

7.4　计数器指令

S7-300 PLC 为计数器保留了一块计数器存储器区域，每个计数器保留了一个 16 位的字和一个二进制位，其中计数器字用来存放它的当前计数值，计数器位用来保存计数器的状态；采用计数器地址来访问计数器，其编址方式为 C 加编号，如 C1、C100 等。不同的 CPU 模板，用于计数器的存储区域不同，最多允许使用 64~512 个计数器。计数器指令可分为加计数器（S_CU）、减计数器（S_CD）、加/减计数器（S_CUD）及计数器线圈指令。各计数器只要当前计数值不为 0，其动合触点状态均为 1，输出端 Q 状态均为 1；在同一段程序中，各种计数器地址不能重复使用。

7.4.1　加计数器（S_CU）

加计数器（S_CU）指令框如图 7-42 所示，CU 为加计数器的输入端，S 为加计数器的设置端，PV 为计数器的预置输入端，R 为计数器的复位输入端，Q 为计数器位输出端，CV、CV_BCD 为计数器当前值的两种表示格式，分别为十六进制和 BCD 码表达式。

图 7-42a 为加计数器原始状态，计数器当前值为 0，输出端 Q 状态为 0，计数器动合触点 C0 也为 0；如图 7-42b 所示，计数端 CU 接 I0.0 信号且上升沿有效，当 I0.0 状态由 0 变为 1 时，计数器当前值为 1，输出端 Q 状态为 1，动合触点 C0 为 1；如图 7-42c 所示，当 S 端有效（M0.0 上升沿有效）时，无论当前值为何值，当前值都为预设置；如图 7-42d 所示，无论当前值为何值，只要复位端信号为 1，则计数器被清 0，Q 端输出为 0，计数器动合触点 C0 也为 0。

在 S 端置位时，如果加计数器输入端 CU 状态为 1，则下一个扫描周期在置位值的基础上直接加 1；如果计数器当前值小于 999，则 CU 端只要输入一个上升沿，计数器当前值就加 1。

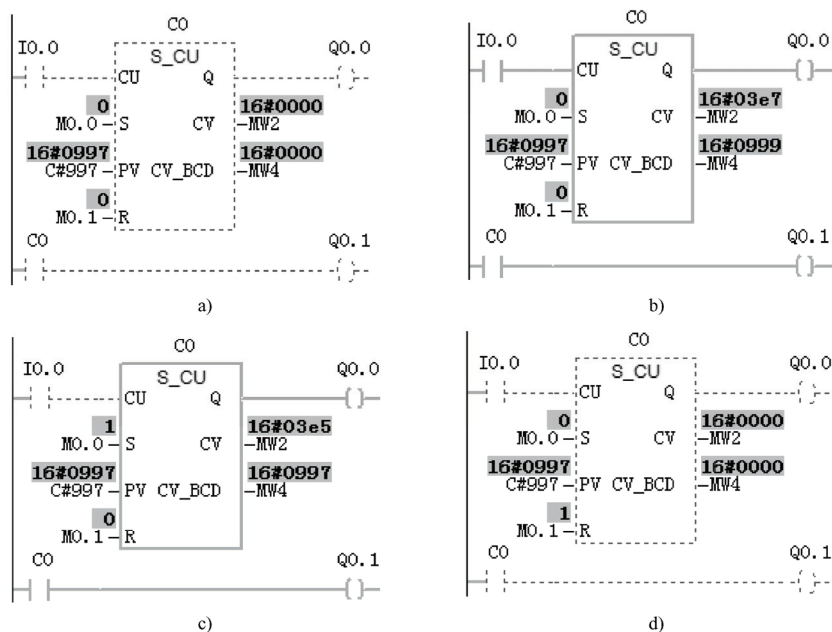


图 7-42　加计数器性能

a) 加计数器梯形图　b) I0.0 上升沿时，计数器当前值加 1　c) M0.0 为 1 时，则当前值为预设值　d) M0.1 为 1 时，计数器复位

7.4.2　减计数器（S_CD）

减计数器(S_CD)指令框如图 7-43 所示，CD 为减计数器的输入端，S 为减计数器的设置端，PV 为减计数器的预置输入端，R 为减计数器的复位输入端，Q 为减计数器的位输出端，CV、CV_BCD 为计数器当前值的两种表示格式，分别为十六进制和 BCD 码表达式。

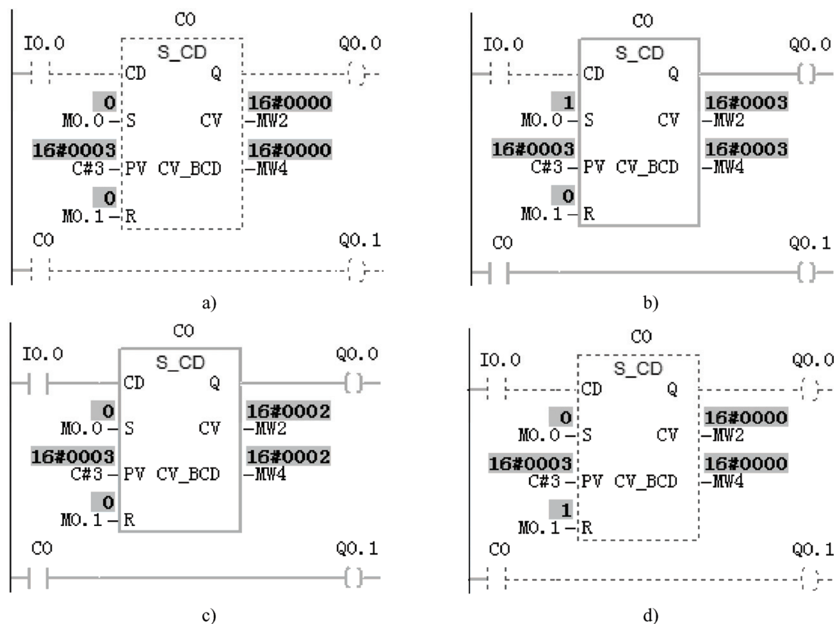


图 7-43　减计数器性能

a) 减计数器梯形图　b) M0.0 为 1 时，则当前值为 3　c) I0.0 上升沿时，计数器当前值减 1　d) M0.1 为 1 时，计数器复位

图 7-43a 为减计数器原始状态，计数器当前值为 0，输出端 Q 状态为 0，计数器动合触点 C0 也为 0，如果当前值为 0，则即使减计数器计数端有上升沿输入，计数器当前值仍然为 0，输出端 Q 及动合触点 C0 状态仍保持为 0；如图 7-43b 所示，当 S 端有效（M0.0 上升沿有效）时，无论当前值为何值，当前值都为预设置；如图 7-43c 所示，计数端 CD 接外部计数信号 I0.0，当 I0.0 状态由 0 变为 1 时，计数器当前值减 1；如图 7-43d 所示，无论当前值为何值，只要复位端为 1，则计数值被清 0，当前值为 0，Q 端输出为 0，计数器动合触点 C0 也为 0。

在 S 端置位时，如果减计数器输入端 CD 状态为 1，则下一个扫描周期在置位值的基础上直接减 1；如果计数器当前值等于 0，则 CD 端即使输入一个上升沿，计数器当前值也不再变化。

7.4.3 加减计数器（S_CUD）

加减计数器（S_CUD）指令框如图 7-44 所示，CU 为加计数器的输入端，CD 为减计数器的输入端，S 为加减计数器的设置端，PV 为计数器的预置输入端，R 为计数器的复位输入端，Q 为计数器位输出端，CV、CV_BCD 为计数器当前值的两种表示格式，分别为十六进制和 BCD 码表达式。

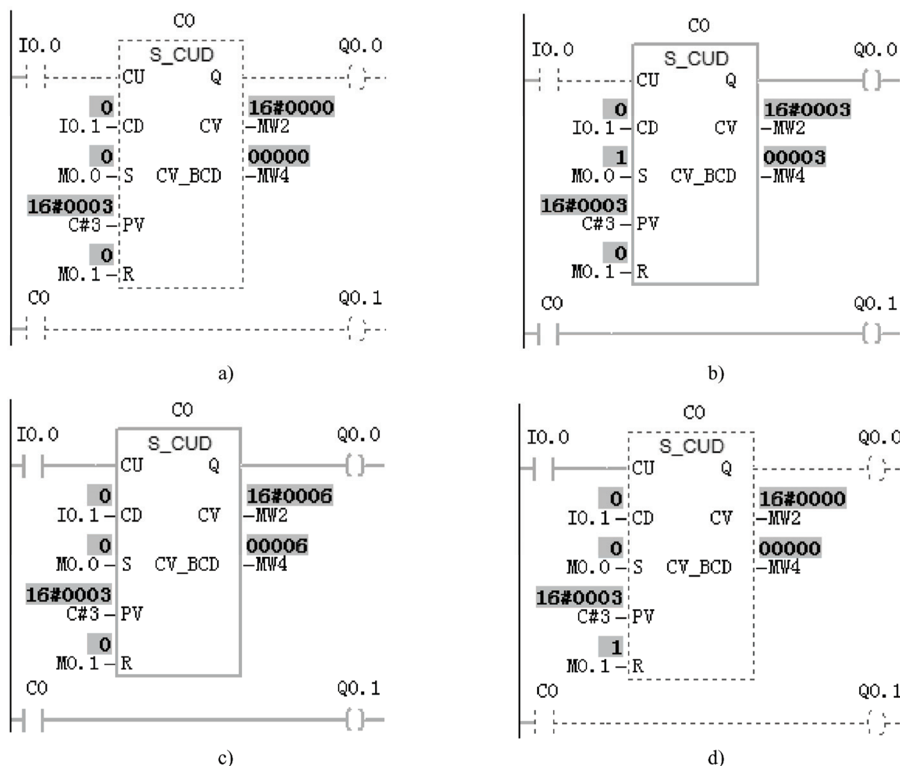


图 7-44　加减计数器性能

a) 加减计数器梯形图　b) M0.0 为 1 时，则当前值为 3　c) I0.0/I0.1 上升沿时，计数器当前值变化　d) M0.1 为 1 时，计数器复位

图 7-44a 为加减计数器原始状态，计数器当前值为 0，输出端 Q 状态为 0，计数器动合触点 C0 也为 0；如图 7-44b 所示，当 S 端有效（M0.0 上升沿有效）时，无论当前值为何值，当前值都为预设置；如图 7-44c 所示，输入端 CU 接 I0.0 信号且上升沿有效时，计数器当前值加 1；输入端 CD 接 I0.1 信号，当 I0.1 状态由 0 变为 1 时，计数器当前值减 1；如图 7-44d 所示，无论当前值为何值，只要复位端为 1，则计数值被清 0，当前值为 0，Q 端输出为 0，计数器动合触点 C0 也为 0。

在 S 端置位时，如果加计数器输入端 CU 状态为 1，则下一个扫描周期在置位值的基础上直接加 1；如果减计数器输入端 CD 状态为 1，则下一个扫描周期在置位值的基础上直接减 1；如果两个计数输入端均为上升沿，两条指令均被执行，当前值保持不变。

7.4.4 计数器的线圈指令

除了前面介绍的块图形式的计数器指令以外，S7-300 系统还为用户准备了 LAD 环境下的线圈形式的计数器，指令包括计数器初值预置指令 SC、加计数器指令 CU 和减计数器指令 CD。

加计数器指令 CU 和减计数器指令 CD 需要与计数器初值预置指令 SC 配合使用，分别可以实现 S_CU 与 S_CD 指令的功能，应用举例如图 7-45 所示。

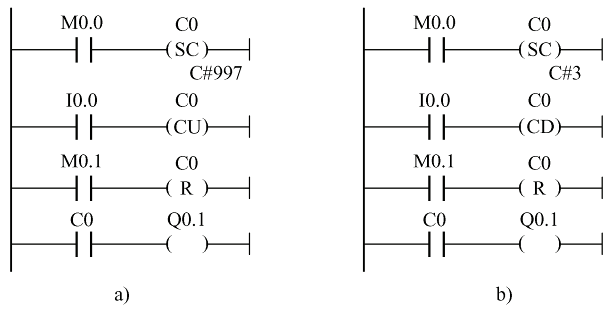


图 7-45 计数器线圈指令应用

a) 加计数器线圈指令 b) 减计数器线圈指令

7.5 CPU 时钟存储器

S7-300 CPU 本身带有时钟存储器功能。要使用该项功能，在硬件配置时需要设置 CPU 的属性，如图 7-46 所示，在 “周期/时钟存储器”选项卡中，勾选“时钟存储器”复选框就可激活存储器时钟功能。

时钟存储器的字节地址可以自行更改，在“时钟存储器”区域输入想为该项功能设置的 MB 的地址，如需要使用 MB100，则直接输入 100。“时钟存储器”的功能是对所定义的 MB 的各个位周期性地（占空比为 1∶1）改变其二进制值，存储器位对应的周期及频率见表 7-9。

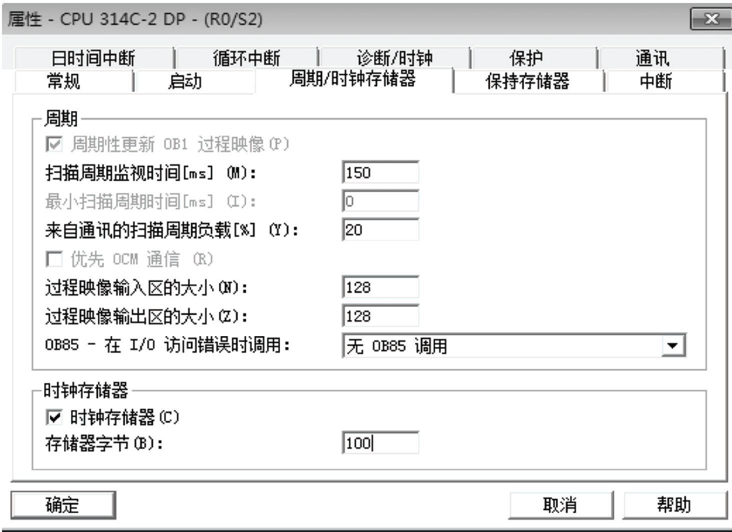


图 7-46　存储器时钟功能设置

表 7-9　时钟存储器

字节位序	7	6	5	4	3	2	1	0
周期/s	2	1.6	1	0.8	0.5	0.4	0.2	0.1
频率/Hz	0.5	0.625	1	1.25	2	2.5	5	10

【例 7-8】 当按下按钮 SB1（I0.0）时，指示灯（Q0.0）以 0.5Hz 频率闪烁；当按下按钮 SB2（I0.1）时，指示灯（Q0.1）以 2Hz 频率闪烁，按下停止按钮 SB2（I0.3），指示灯熄灭。

梯形图设计如图 7-47 所示。

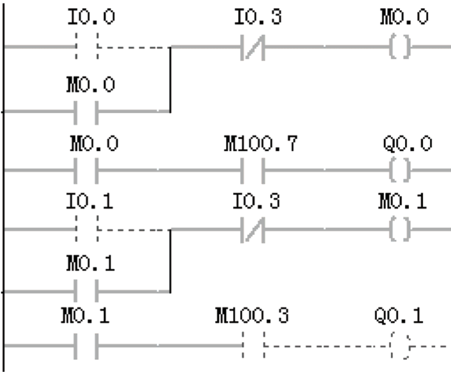


图 7-47　时钟存储器应用示例

分析：在图 7-47 中将字节 MB100 设置为时钟存储器，每一位的振荡周期见表 7-9。当 CPU 运行时，M100.7 触点按照 1s 接通、1s 断开的周期运行着，当按下启动按钮 SB1(I0.0)时，M0.0 得电并自锁，使得 Q0.0 按照 M100.7 触点通断的节奏运行。同理，可分析 Q0.1 的运行情况。

7.6 数据处理指令

7.6.1 传送指令

S7-300 PLC 梯形图的传送指令只有 MOVE 一条指令，它将源数据直接传送到目的地址，指令框如图 7−48 所示。输入变量 IN 和输出变量 OUT 可以是 8 位、16 位或 32 位的基本数据类型。

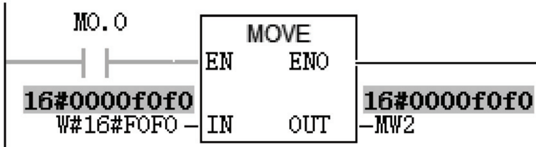


图 7−48 MOVE 指令框

输入端与输出端的数据可以相同也可以不同。从图 7−49 的变量表可以看出，当（MW2）内容传递给（MB4）时，只将字数据的低 8 位传递给了 MB4；当（MW2）内容传递给（MD6）时，由于传递数据的位数不足，MD6 的高 16 位补 0。

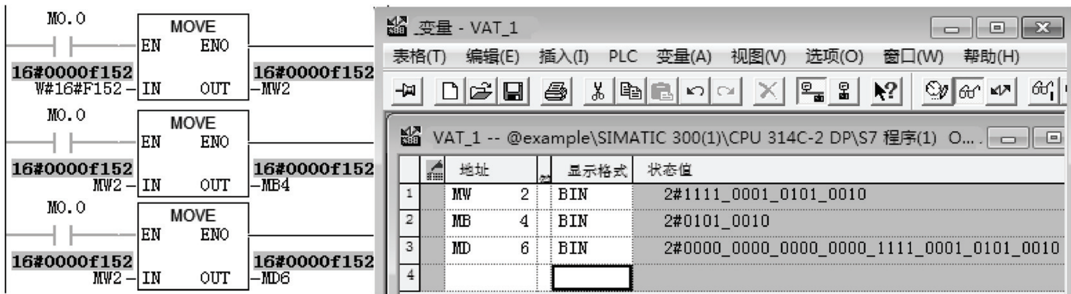


图 7−49 MOVE 指令应用

7.6.2 比较指令

比较指令用于比较两个数据的大小，如果比较结果为“真”，则函数的 RLO 为“1”，否则为“0”，指令框格式如图 7−50 所示，图中 CMP 后面内容为比较类型，INT1 为比较数（S1），INT2 为被比较数（S2）。该指令的使用方法与标准触点类似，可位于任何放置标准触点的位置，并可根据需要选择比较的数据类型及比较的形式类型，见表 7−10。

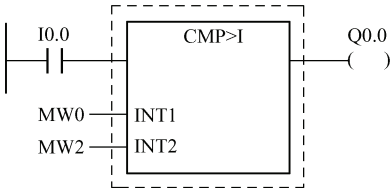


图 7−50 比较指令

表 7-10　比较类型

指令	指令功能		指令	指令功能	
EQ_I	整数	(S1)＝(S2) 时，RLO=1	LT_D	双整数	(S1)<(S2) 时，RLO=1
NE_I		(S1)≠(S2) 时，RLO=1	GE_D		(S1)≥(S2) 时，RLO=1
GT_I		(S1)>(S2) 时，RLO=1	LE_D		(S1)≤(S2) 时，RLO=1
LT_I		(S1)<(S2) 时，RLO=1	EQ_R	实数	(S1)＝(S2) 时，RLO=1
GE_I		(S1)≥(S2) 时，RLO=1	NE_R		(S1)≠(S2) 时，RLO=1
LE_I		(S1)≤(S2) 时，RLO=1	GT_R		(S1)>(S2) 时，RLO=1
EQ_D	双整数	(S1)＝(S2) 时，RLO=1	LT_R		(S1)<(S2) 时，RLO=1
NE_D		(S1)≠(S2) 时，RLO=1	GE_R		(S1)≥(S2) 时，RLO=1
GT_D		(S1)>(S2) 时，RLO=1	LE_R		(S1)≤(S2) 时，RLO=1

【例 7-9】 比较指令的应用。

分析：比较指令框的输入/输出均为 BOOL 变量，当输入条件满足时，指令框工作，当比较条件满足时，指令框输出状态为 1。在图 7-51a 中，当两个被比较的整数值不相等时（<>=），指令条件满足，Q0.0=1；在图 7-51b 中，当两个双整数相等时（= =D），指令条件满足，Q0.1=1；在图 7-51c 中，当第一个浮点数小于等于第二个浮点数时（< =R），指令条件满足，Q0.2=1；在图 7-51d 中，当第一个浮点数大于第二个浮点数时（1224>1223），指令条件不满足，Q0.2=0。

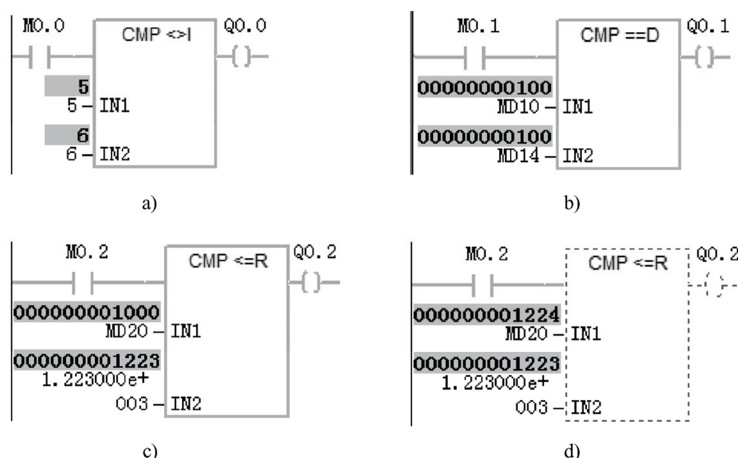


图 7-51　比较指令应用

7.6.3　数据转换指令

转换指令是将累加器 1 中的数据进行数据类型转换，转换结果仍存放在累加器 1 中，可以实现 BCD 码与整数、整数与长整数、长整数与实数、整数的反码、整数的补码以及实数求反等数据转换操作，转换类型见表 7-11。

表 7-11　转换类型

指　　令	指令功能	
BCD_I	BCD 与整数的转换指令	将 BCD 码转换成整数
BCD_DI		将 BCD 码转换成双整数
I_BCD		将整数转换成 BCD 码
DI_BCD		将双整数转换成 BCD 码
I_DI	整数转换指令	将整数转换成双整数
DI_R		将双整数转换成浮点数
INV_I	取反指令	对整数二进制取反
INV_DI		对双整数二进制取反
NEG_I	补码指令	对整数求补码
NEG_DI		对长整数求补码
NEG_R		对浮点数取反
ROUND	取整指令	将 32 位 IEEE 浮点数转换为 32 位整型数，并将结果取整为最接近的整数
TRUNC		将 32 位 IEEE 浮点数转换为 32 位整型数，运算结果为所转换浮点数的整数部分
CEIL		将 32 位 IEEE 浮点数转换为 32 位整型数，并将运算结果取整为大于或等于转换后数字的最小整数
FLOOR		将 32 位 IEEE 浮点数转换为 32 位整型数，并将运算结果取整为小于或等于所转换浮点数的最大整数

【例 7-10】 图 7-52 为梯形图在线运行监控情况，观察数据转换情况并分析数据转换关系。

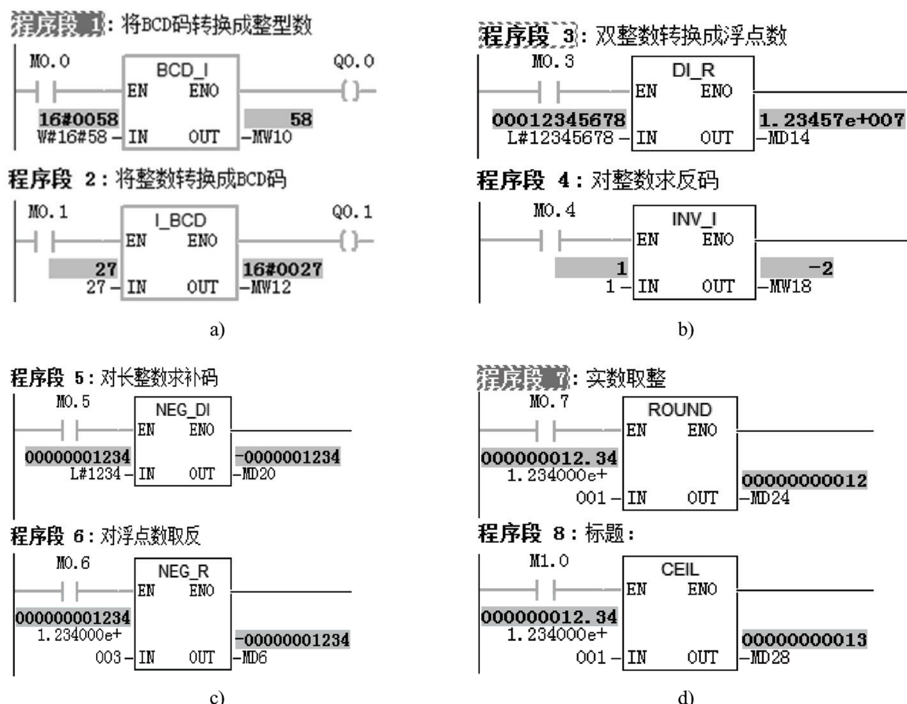


图 7-52　数据类型转换应用示例

分析：在图 7−52a 中，程序段 1 将 3 位 BCD 码转换成整数，程序段 2 将整数转换为 3 位 BCD 码；在图 7−52b 中，程序段 3 将长整数转换为 32 位的浮点数，程序段 4 对整型数逐位求反，结果保存在累加器 1 的低字中，并传送到 MW18 中；在图 7−52c 中，程序段 5 求长整数的二进制补码，程序段 6 对累加器 1 中的 32 位浮点数求反，相当于乘以（−1）；在图 7−52d 中，程序段 7 将浮点数四舍五入取整，程序段 8 对浮点数取大于或等于该数的整数。

7.7　数学运算指令

数学运算指令可完成整数、长整数及实数的加、减、乘、除、求余和求绝对值等基本算数运算，32 位浮点数的平方、平方根、自然对数、基于 e 的指数运算及三角函数等扩展算数运算，以及数据的移位和逻辑运算。数学运算指令见表 7−12。

表 7−12　数学运算指令类型

指　　令	含　　义	指　　令	含　　义
ADD_I	整数加/减/乘/除运算	ABS	求浮点数的绝对值
SUB_I		SQRT	求浮点数的平方根
MUL_I		SQR	求浮点数的平方
DIV_I		LN	求浮点数的自然对数
ADD_DI	双整数加/减/乘/除运算	EXP	对浮点数求以 e 为底的指数值
SUB_DI		SIN	求浮点数（弧度）的正弦值
MUL_DI		COS	求浮点数（弧度）的余弦值
DIV_DI		TAN	求浮点数（弧度）的正切值
ADD_R	浮点数加/减/乘/除运算	ASIN	求[−1,1]范围内浮点数的反正弦值
SUB_R		ACOS	求[−1,1]范围内浮点数的反余弦值
MUL_R		ATAN	求浮点数的反正切值
DIV_R		MOD_DI	返回长整型数除法的余数
SHR_I	（整数/双整数）右移	WAND_W	（单/双字）与运算
SHR_DI		WAND_DW	
SHL_W	（单/双字）左移	WOR_W	（单/双字）或运算
SHL_DW		WOR_DW	
SHR_W	（单/双字）右移	WXOR_W	（单/双字）异或运算
SHR_DW		WXOR_DW	
ROL_DW	（双字）左循环移位指令	ROR_DW	（双字）右循环移位指令

【例 7−11】 整数运算指令的应用。

分析：在图 7−53 中，两个整数相加（5+27），结果 32 存放在 MW2 中；若 M0.1=1，则将 16#FE 传送到被除数 MD4 中，则双整数运算结果（MD4）÷ 5 放入 MD8 寄存器中，即（MD8）=50；若程序刚开始运行，且 M0.1 = 0，则 MD4 的初始值是 0，由于 0 除任何数都等于 0，所以（MD8）=0。

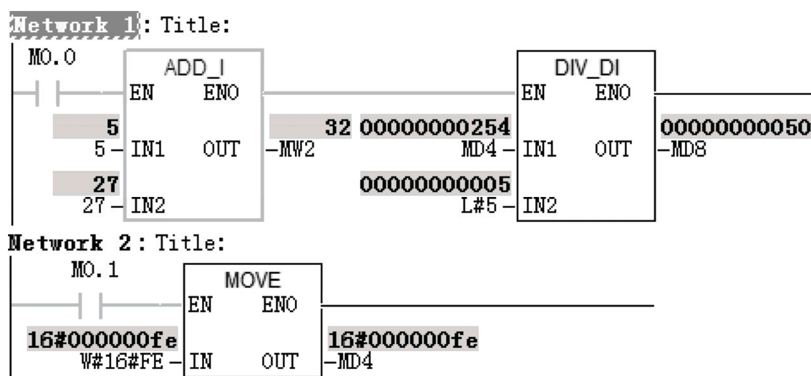


图 7-53　整数运算指令应用示例

【例 7-12】 浮点数运算指令应用 1。

分析：在图 7-54 中，MUL_R 为实数相乘指令，实数 2*3.125 = 6.25，结果存入 MD2 寄存器中；SQRT 为求实数的平方根指令，（9.23）$^{0.5}$ ≈ 3.03809，结果存入 MD6 寄存器中。

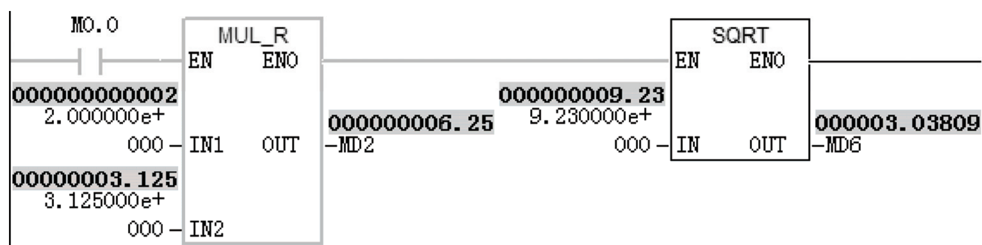


图 7-54　浮点数运算指令应用示例 1

【例 7-13】 浮点数运算指令应用 2。

分析：观察图 7-55。在网络 1 中，MOD_DI 指令对长整型求余数，（49/5）的余数是 4，结果存入 MD2 寄存器中；ABS 指令对浮点数取绝对值，｜-5.2｜= 5.2，结果存入 MD6 寄存器中；在网络 2 中，LN 指令用来求浮点数的自然对数，ln10 = 2.30259，结果存入 MD10 寄存器中；EXP 指令用来求浮点数以 e 为底的指数值，$e^{2.30259}$=10，结果存入 MD14 寄存器中。

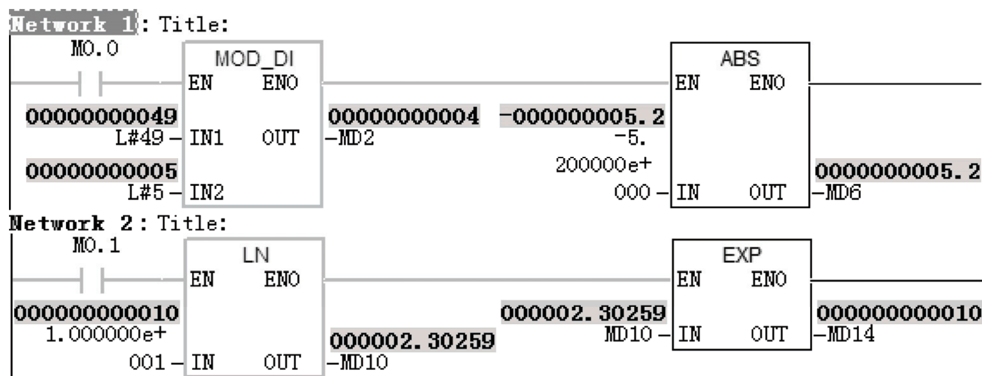


图 7-55　浮点数运算指令应用示例 2

【例 7-14】 三角函数运算指令应用。

分析：观察图 7-56，SIN 指令对浮点数（弧度）求正弦值，（π/2）=1.570796，sin（（π/2）=1，结果存入 MD2 寄存器中；ATAN 指令对浮点数求反正切值，arctan(−1) = (− π/4) = 0.785398，结果存入 MD6 寄存器中。

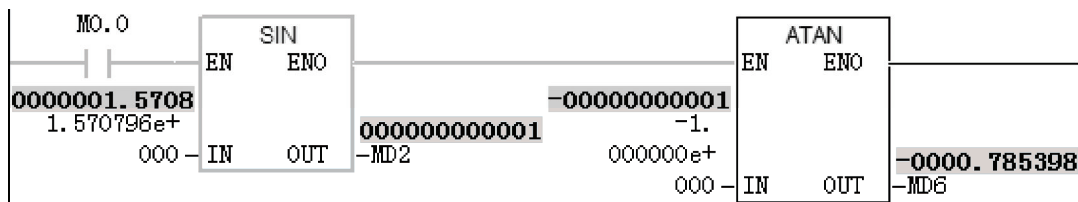


图 7-56 浮点数运算指令应用示例

【例 7-15】 移位指令的应用。

分析：观察图 7-57。网络 1 用于给变量 MW2、MW12 赋初值；网络 2 和网络 3 中，SHR_I/SHR_DI 指令用于整数/双整数右移、SHL_W/SHL_DW 指令用于对字/双字左移，从而实现移位功能，其中输入 N 用于指定移位的位数。数据移动 1 次后的变化如图 7-58 所示。

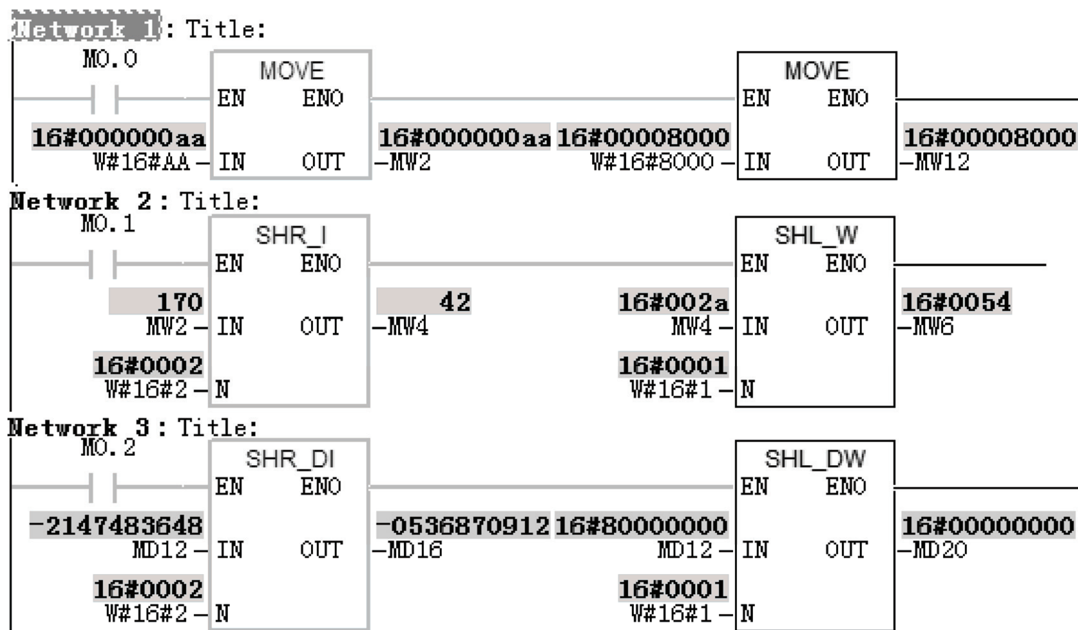


图 7-57 移位指令应用示例

从图 7-58 的标注 1 和标注 2 数据移位可见，无论是右移还是左移，对于整数为正而言，都是给空出位补 N 个“0”；从标注 3 数据移位可见，整数为负且右移时，给空出位补 N 个“1”。

Var - [VAT_1 -- @777\SIMATIC 300(1)\CPU 313C\S7 Program(1) ONLINE]

Table Edit Insert PLC Variable View Options Window Help

	Address	Symbol	Display format	Status value
1	MW 2		HEX	W#16#00AA
2	MW 2		DEC	170
3	MW 2		BIN	2#0000_0000_1010_1010
4	MW 4		BIN	2#0000_0000_0010_1010
5	MW 6		BIN	2#0000_0000_0101_0100
6	MD 12		BIN	2#1000_0000_0000_0000_0000_0000_0000_0000
7	MD 16		BIN	2#1110_0000_0000_0000_0000_0000_0000_0000
8	MD 20		BIN	2#0000_0000_0000_0000_0000_0000_0000_0000

图 7-58 移位一次后的数据对比

【例 7-16】 循环移位指令应用。

分析：观察图 7-59。网络 1 用于给变量 MD2、MD12 赋初值；网络 2 为循环左/右移位指令，用于将输入 IN 的所有内容向左或向右逐位循环移动，N 为移动位数，移位原理如图 7-60 所示，图 7-61 为图 7-59 移动一次后的数据在线监控。

Network 1: Title:
M0.0
MOVE EN ENO
16#f0000000 DW#16#F0000000 - IN OUT - MD2 16#f0000000
MOVE EN ENO
16#80000001 DW#16#80000001 - IN OUT - MD12 16#80000001

Network 2: Title:
M0.1
ROL_DW EN ENO
16#f0000000 MD2 - IN OUT - MD6 16#c0000003
16#0002 W#16#2 - N
ROR_DW EN ENO
16#80000001 MD12 - IN OUT - MD16 16#c0000000
16#0001 W#16#1 - N

图 7-59 循环移位指令应用示例

←左循环

→右循环

图 7-60 移位指令性能

Var - [VAT_1 -- @777\SIMATIC 300(1)\CPU 313C\S7 Program(1) ONLINE]

Table Edit Insert PLC Variable View Options Window Help

	Address	Symbol	Display format		Status value
1	MD 2		BIN	1	2#1111_0000_0000_0000_0000_0000_0000_0000
2	MD 6		BIN		2#1100_0000_0000_0000_0000_0000_0000_0011
3	MW 12		BIN		2#1000_0000_0000_0000
4	MD 12		BIN	2	2#1000_0000_0000_0000_0000_0000_0000_0001
5	MD 16		BIN		2#1100_0000_0000_0000_0000_0000_0000_0000

图 7-61　移动一次后的数据对比

7.8　控制指令

控制指令用于编写结构化程序、优化控制程序结构，以便减少程序执行时间，包括逻辑控制指令和程序控制指令。

7.8.1　逻辑控制指令

可以在组织块（OB）、功能块（FB）和功能（FC）中使用逻辑控制指令，包括跳转（JMP）指令和若“否”则跳转（JMPN）指令，每个指令还必须有与之对应的目标（LABEL）。跳转指令和标号间的所有指令都不予执行。

逻辑控制指令应用如图 7-62 和图 7-63 所示，其中跳转指令的标号最多包含四个字符，第一个字符必须是字母表中的字母，其他字符可以是字母或数字。

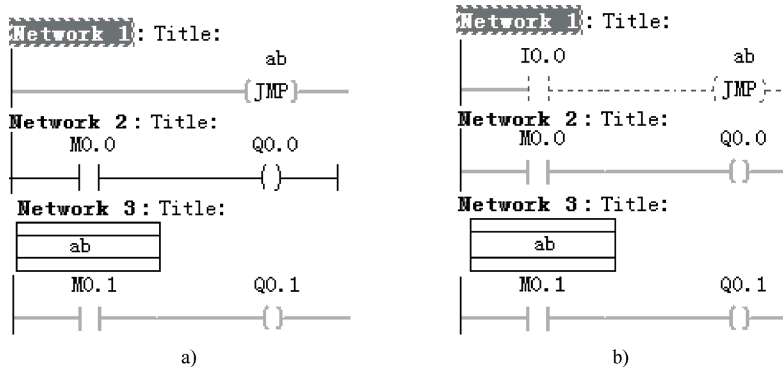


图 7-62　JMP 指令应用

a) 无条件跳转　b) 有条件跳转

图 7-62a 的跳转是无条件跳转，程序始终执行跳转，并忽略跳转指令与跳转标号之间的指令，即使 M0.0 的状态为 1，Q0.0 状态也始终为 0。图 7-62b 的跳转是有条件跳转，当跳转条件 I0.0 不满足时，程序顺序执行；跳转条件 I0.0=1 时，则跳转到标号 ab 处，由于跳转条件

的存在，即使 M0.0 改变为 0，也不会再执行网络 2 的逻辑，即 Q0.0 仍然保持 1 的状态。

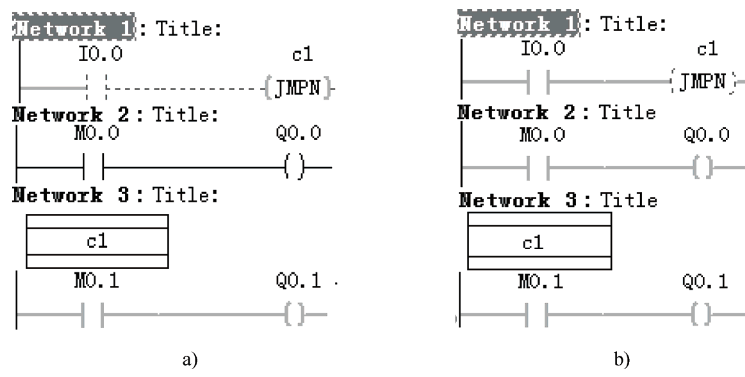


图 7-63　JMPN 指令应用

a) I0.0=0 跳转　b) I0.0=1 不跳转

图 7-63a 是若跳转条件为“否”，则执行跳转，忽略跳转指令与跳转标号之间的指令，即使 M0.0 的状态为 1，Q0.0 状态也始终为 0；图 7-63b 是跳转条件 I0.0=1，所以跳转条件不满足，程序顺序执行，当 M0.0 的状态为 1 时，Q0.0 状态也为 1。

7.8.2　程序控制指令

程序控制指令是指功能块（FB、FC、SFB、SFC）调用指令和逻辑块（OB，FB，FC）结束指令。调用块或结束块可以是有条件的或无条件的。STEP 7 编程软件提供了 6 条程序控制指令，见表 7-13。

表 7-13　主控继电器指令

STL 指令	LAD 指令	说　明
UC	----(CALL)	用于调用没有传递参数的功能（FC）或系统功能（SFC）
MCRA	----(MCRA)	激活 MCR 区，表明一个 MCR 区域的开始
MCRD	----(MCRD)	表明一个按 MCR 方式操作区域的结束
)MCR	----(MCR<)	主控继电器激活
MCR(	----(MCR>)	主控继电器取消激活
BEC	----RET	有条件退出块。若 RLO=1，则结束当前块的扫描，将控制返回给调用块；若 RLO=0，则将 RLO 置 1，程序继续在当前块内扫描

7.9　逻辑块

逻辑块包括组织块（OB）、功能（FC）和功能块（FB），也为程序块。

7.9.1　组织块及其应用

组织块是由操作系统直接调用的逻辑块，是 CPU 的操作系统与用户程序之间的接口，如图 7-64 所示。组织块不能互相调用，其基本功能是调用用户程序。

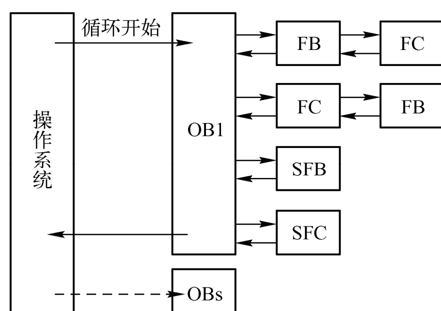


图 7-64　组织块与操作系统关系

组织块可分为循环执行组织块（OB1）、启动组织块（OB100，OB101，OB102）、定期执行组织块（OB10~OB17，OB20~OB23，OB30~OB38，OB40~OB47）、事件驱动组织块（OB80~OB87，OB121~OB122）及背景组织块。

组织块由变量声明表和用户程序组成，各个组织块（除 OB1）实质上是用于各种中断处理的中断服务程序。每个组织块均有优先级，通常情况下组织号码越大，优先级越高。

1．循环执行组织块

在 OB1 中的用户程序是循环执行的主程序。例如前面一直使用的主程序即循环执行组织块 OB1，优先级最低为 1，其循环时间被监控，除 OB90 以外其他所有 OB 均可中断 OB1 的执行。

以下两个事件可导致操作系统调用 OB1：

1）CPU 启动完毕。

2）OB1 执行到上一个循环周期结束。

2．启动组织块

OB100 为暖启动，OB101 为热启动（S7-400 用），OB102 为冷启动。

暖启动：完全再启动。启动时过程映像和不保持的标志存储器、定时器及计数器被清除，保持的标志存储器、定时器和计数器以及数据块的当前值保持（仅当有后备电池，如果使用 EPROM 并且 CPU 的保持特性已赋参数时，S7-300 甚至没有后备电池）。CPU 会自动调用 OB100 一次，然后循环程序 OB1 开始执行。

冷启动：CPU318-2 还具有冷启动型的启动方式。针对电源故障可以定义这种启动方式。它是通过硬件组态时的 CPU 参数来设置。冷启动时，所有过程映像和标志存储器、定时器和计数器被清除（甚至保持的），而且数据块的当前值被装载存储器的当前值（即原来下装到 CPU 的数据块）覆盖。 CPU 会自动调用 OB102 一次，然后循环程序 OB1 开始执行。

对于 OB100~OB102，CPU 只在启动运行时对其进行一次扫描，其他时间只对 OB1 进行循环扫描。

【例 7-17】 开机后系统自动检测三个传感器回路是否完好。如果完好，则相应指示灯点亮，同时蜂鸣器发出响声 1s，之后将灯全部熄灭，等待系统发出下一步动作指令；否则，蜂鸣器不响，即检测不通过。

1）建立工程界面如图 7-65 所示。

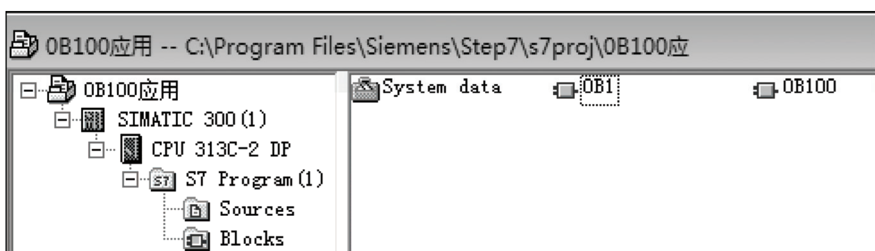


图 7-65　OB100 的建立

2）编写程序。OB100 启动程序编写如图 7-66 所示，OB1 程序编写如图 7-67 所示。

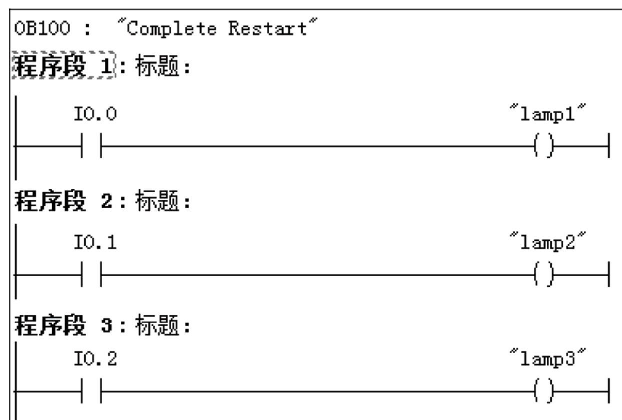


图 7-66　OB100 程序

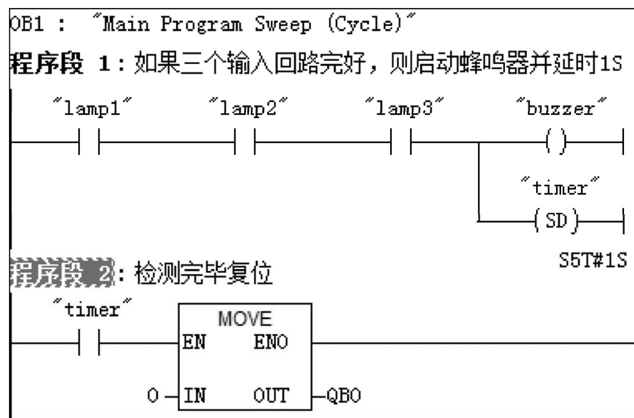


图 7-67　OB1 程序

系统启动运行时，CPU 先执行 OB100，再进入 OB1 循环程序，当传感器回路检测完毕后，将故障指示灯复位，只有当 CPU 再次执行热启动后才会再次执行 OB100 检测回路。

3．定期执行组织块

定期执行组织块包括日期时间中断组织块、延时中断组织块、循环中断组织块及硬件中

断组织块。

日期时间中断组织块（OB10~OB17）用来设置触发的日期、执行模式，使其在某一特定日期或特定间隔产生中断，8 个日期时间中断具有相同的优先级。

延时中断组织块（OB20~OB23）延时时间为 1~60000ms，精度为 1ms，优先级及更新过程映像区的参数在 CPU 的属性中设置；SFC32（SRT_DINT）设置 OB 号与延时时间，触发中断；SFC32（CAN_DINT）取消延时中断；SFC33（QRY_DINT）查询延时中断的状态。

循环中断组织块（OB30~OB38）的循环中断间隔在 CPU 的属性中设置，每个 OB 默认时间间隔不同，系统功能 SFC39~SFC42 可以禁止、延迟、使能循环中断。

硬件中断组织块（OB40~OB47）也称为过程中断，由外部设备产生，SFC39~SFC42 可以禁止、延迟、使能硬件中断。

【例 7-18】 使用 OB10 组织块完成每天下班时间打铃 2s 的功能。

1）新建工程 EX_OB10，组态硬件，并双击 CPU，选择“Time-of-Day Interrupts”选项卡，可以看到 CPU 支持 OB10，如图 7-68 所示。

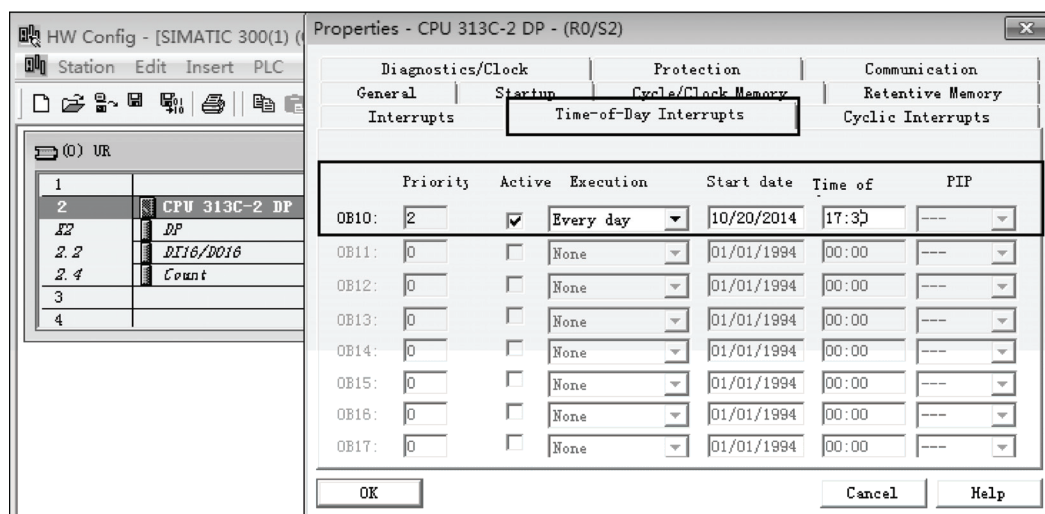


图 7-68　Time-of-Day Interrupts 选项内容

选中“Active”→设置“Execution”选项，该选项下拉菜单可选择“None/Once/Every minute/Every hour/Every week/Every month/End of month/Every year”→设置开始执行的日期和时间。

硬件组态完毕后编译并保存硬件组态文件。

2）OB10 程序设计。每隔 24h，系统访问 OB10 一次，OB10 的任务是下班时间到就启动打铃系统，程序如图 7-69 所示。

3）OB1 程序设计。当系统访问 OB10 后，Q0.0=1，启动打铃装置，铃声持续 2s 后断开，OB1 实现功能如图 7-70 所示；当 24h 后，系统再次访问 OB10，使得 Q0.0=1，如此循环下去，完成在每天下班时间 17：30 时启动系统打铃 2s 的功能。



图 7-69　OB10 程序　　　　图 7-70　OB1 程序

4）将硬件组态、OB1、OB10 分别下载到 CPU 中并进行系统联调。

【例 7-19】 使用 OB20 延时中断，每隔 1s 计数 1 次，计数满 5 个后发出打包指令，如果检测到次品则不计数，周而复始。

1）新建工程 EX_OB20，组态硬件，并双击 CPU，选择 Interrupts 选项卡，可以看到 CPU 支持 OB20，如图 7-71 所示，硬件组态完毕后编译并保存硬件组态文件。

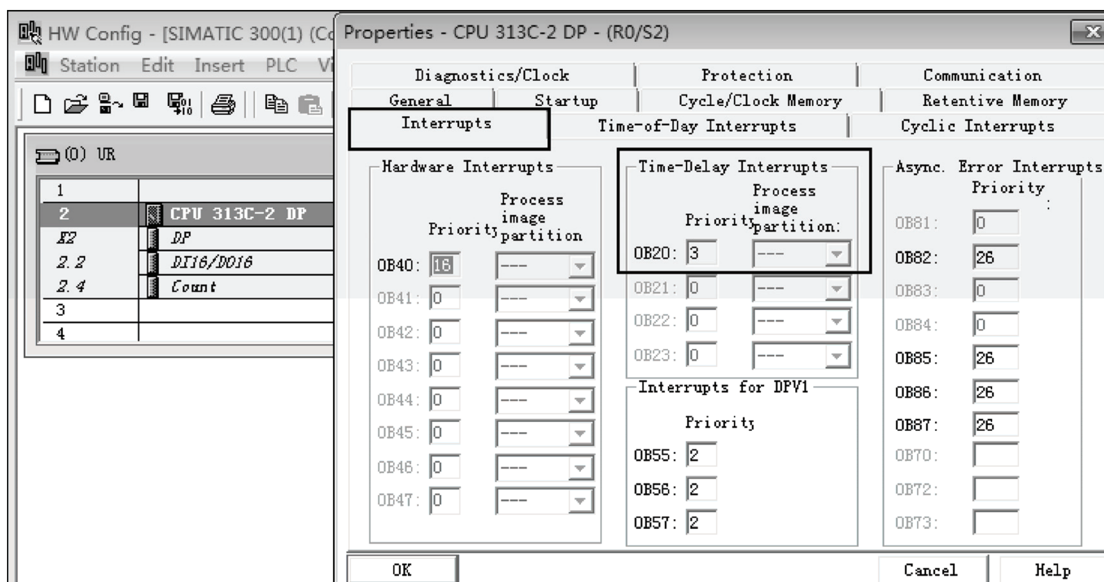


图 7-71　Interrupts 选项卡内容

变量分配见表 7-14。

表 7-14　变量分配表

S7 Program(1) (Symbols) -- EX_OB20\SIMATIC 300(1)\C

	Symbol	Address		Data typ
1	产品检测	I	0.0	BOOL
2	次品检测	I	0.1	BOOL
3	打包指令	Q	0.0	BOOL
4	延时中断状态	MW	8	WORD
5	合格产品计数	MW	100	WORD

2）OB20 程序设计。OB20 为系统延时中断组织块，每隔设定时间，系统访问一次，程序设计如图 7-72 所示。

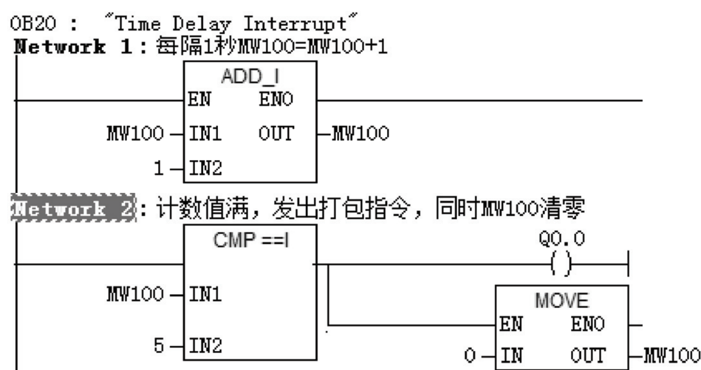


图 7-72　OB20 程序

其中，MW100 为计数值，系统访问一次 OB20，MW100 中的值就加 1 一次，当访问次数满 5 次时，Q0.0 得电，发出打包指令，同时 MW100 清零，重新开始计数。

3）OB1 程序设计。OB1 程序主要用于实现调用延时中断组织块、设定延时时间等功能，程序如图 7-73 所示。

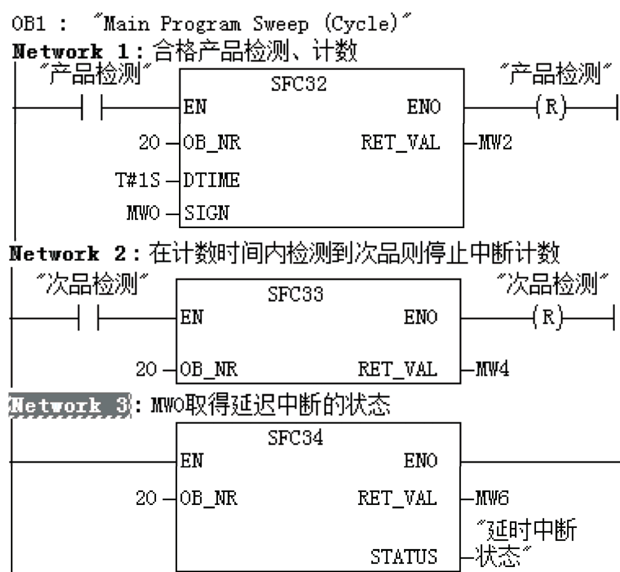


图 7-73　OB1 程序

图 7-73 中网络 1 为调用 OB20 程序。每一次 OB20 的程序执行，必须调用 SFC32（SRT_DINT）系统块。组织块号（20~23）由输入参数 OB_NR=20 给定，延迟时间由输入参数 DTIME=T#1S 给定，REL_VAL 为出错代码；调用 SFC32 且设定的延时时间到，则执行 OB20 程序，如果再次执行 OB20 程序，需要再次调用 SFC32。

图 7-73 中网络 2 为取消延时。如果在延迟时间未到之前想取消程序的执行，可以调用 SFC33（CAN_DINT），同时可以使用 SFC34 取得延迟中断的状态。

图 7-73 中网络 3 为调用 SFC34（QRY_DINT），以便取得延迟中断的状态，状态参数含义见表 7-15。

表 7-15　SFC34 状态位的含义

位	值	含义
0	0	操作系统已启用延时中断
1	0	未拒绝新的延时中断
2	0	未激活延时中断或时间已过
3	—	—
4	0	未加载延时中断
5	0	当前活动的测试功能未禁用延时中断 OB 的执行

4）将硬件组态、OB1、OB20 分别下载到 CPU 中并进行系统联调。

系统联调时，为方便观察系统运行情况，可建立 Variable Table 变量表监控数据实时变化，如图 7-74 所示。

VAT1 -- @EX_OB20\SIMATIC 300(1)\CPU 313C-2 DP\S7 Program(1) ONLINE

	Address	Symbol	Display format	Status value
1	MW 100	"合格产品计数"	DEC	2
2	MW 8	"延时中断状态"	BIN	2#0000_0000_0001_0000
3				

图 7-74　在线监控延时状态

【例 7-20】 使用 OB35 循环中断功能，实现 MW0 每秒加 1 功能，且 MW0=100 时，指示灯亮、MW0 重新开始计数，周而复始。

1）新建工程 EX_OB35，组态硬件，并双击 CPU，选择 Cyclic Interrupts 选项卡，可以看到该款 CPU 支持 OB35，如图 7-75 所示，修改 OB35 执行周期（1~60000ms）为 1000ms，硬件组态完毕后编译并保存硬件组态文件。

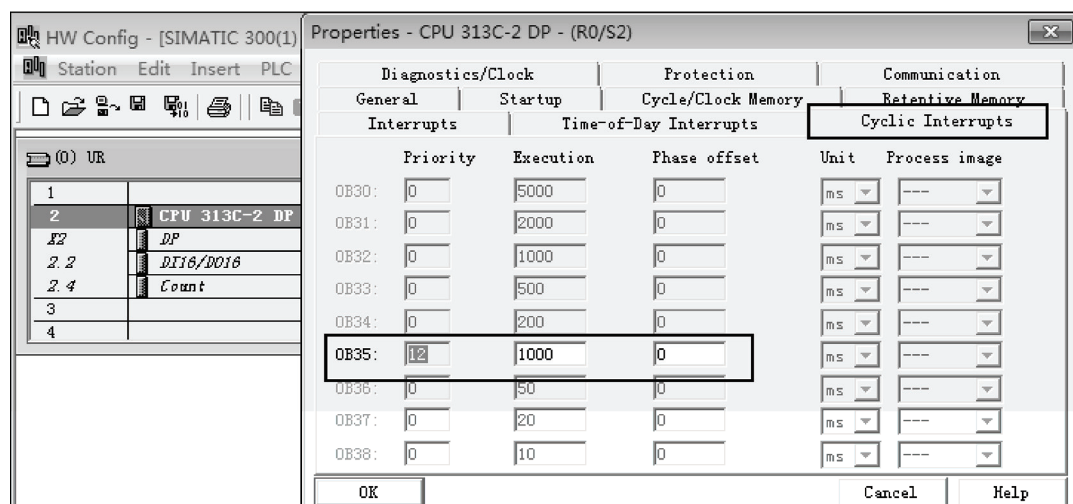


图 7-75　OB35 参数设置

2）OB35 程序设计。在图 7-76 中，系统每隔 1000ms 访问 OB35 模块一次，MW0 = MW0 + 1；当 MW0 = 100 时，计满标志位 M2.0 = 1，同时将 MW0 清零，为下一次重新计数做准备。

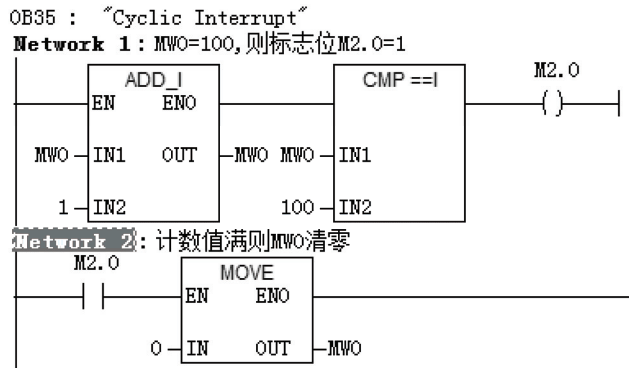


图 7-76　OB35 程序编写

3）OB1 程序设计。OB35 参数已在硬件组态时设置，因此 OB1 程序设计时只需调用 OB35 运行结果，程序设计如图 7-77 所示。

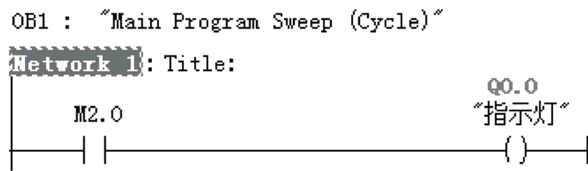


图 7-77　OB1 程序

当 M2.0 = 1 时，Q0.0 = 1，指示灯亮，当指示灯亮 1s 后，OB35 运行、MW0 重新计数，M2.0 导通条件消失，Q0.0 = 0，指示灯熄灭。

4）将硬件组态、OB1、OB35 分别下载到 CPU 中并进行系统联调。

系统联调时，为方便观察系统运行情况，可建立 Variable Table 变量表监控数据实时变化，如图 7-78 所示。

VAT_1 -- @EX_OB35\SIMATIC 300(1)\CPU 313C-2 DP\S7 Progr...

	Address		Symbol	Display format	Status value	Modify value
1	MW	0		DEC	3	
2	M	2.0		BOOL	false	
3	Q	0.0	"指示灯"	BOOL	false	

图 7-78　在线监控 MW0 数据变化情况

【例 7-21】 使用 OB40 循环中断功能，实现当 I0.0 从 0 到 1 的上升沿触发 OB40 一次，触发次数存放在 MW0 中。

1）新建工程 EX_OB40，组态硬件，并双击数字量模块“DI24/DO16”，选择 Inputs 选项卡，如图 7-79 所示，勾选 Hardware interrupt 的 Rising edge 一栏的 I0.0，即选中 I0.0 上升沿硬

件中断，硬件组态完毕后编译并保存硬件组态文件。

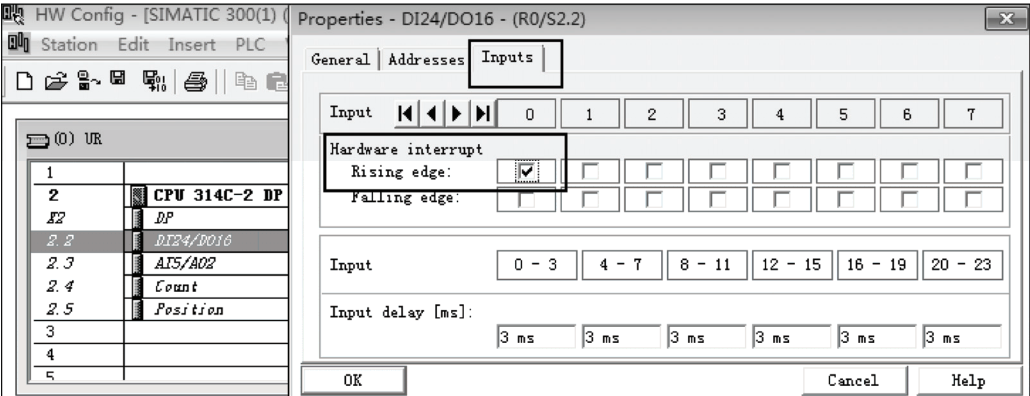


图 7-79　OB40 触发条件设置

2）OB40 程序。如图 7-80 所示，每出现一次 I0.0 由 0 变为 1 的上升沿，系统执行中断程序 OB40 1 次，则 MW0 = MW0 + 1。

3）OB1 程序。为了验证外部按钮信号 I0.0 上升沿触发中断 OB40 程序，在主程序中通过计数器 C0 也记录了 I0.0 出现上升沿的次数，如图 7-81 所示。

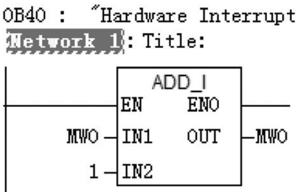


图 7-80　OB40 程序

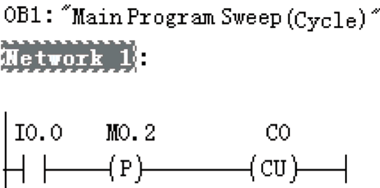


图 7-81　OB1 程序

4）将硬件组态、OB1、OB40 分别下载到 CPU 中并进行系统联调。

系统联调时，为方便观察系统运行情况，可建立 Variable Table 变量表监控数据实时变化，如图 7-82 所示。可见 I0.0 出现多少次上升沿，OB40 就被执行了多少次。

Var - VAT_1

Table Edit Insert PLC Variable View Options Wi

VAT_1 -- @EX_OB40\SIMATIC 300(1)\CPU 314C-2 DP\S

	Address	Symbol	Display format	Status value
1	MW 0		DEC	8
2	C 0		COUNTER	C#8

图 7-82　在线监控硬件中断次数

4．事件驱动组织块

事件驱动组织块及对应事件见表 7-16，可分为异步故障和同步故障两类。

表 7-16　事件驱动组织块及对应事件

OB	特点	错误类型	故障举例	优先级
OB80	异步故障	时间错误	在处理优先级时延时过多	26
OB81		电源故障	未安装后备电池或后备电池失效	26/28
OB82		诊断中断	有诊断功能的模块输入断线	
OB83		移入/移除中断	插入/删除已组态模块	
OB84		CPU 硬件故障	通信总线接口故障	
OB85		程序执行错误	事件触发时响应的 OB 模块未下载	
OB86		机架故障	扩展设备或 DP 主从站故障	
OB87		通信故障	读取信息格式错误	
OB121	同步故障	编程错误	在程序中调用一个 CPU 中并不存在的错误	与被中断的错误 OB 优先级相同
OB122		访问错误	访问一个不存在的模块	

5．背景组织块

背景组织块（OB90）可以保证 CPU 的最短扫描时间，对运行时间要求不高的操作，放在 OB90 中执行，避免出现等待时间。OB90 的优先级最低，其运行时间不受 CPU 操作系统的监视，在 OB90 中编写的程序长度不受限制。

7.9.2　FC 及其应用

FC 是功能，相当于子程序或函数，不具备自己的存储区，可以在共享 DB 里定义数据。如果某项功能多处可以用到，则将其进行功能化编程，在 OB1 或其他功能/功能块中调用，不仅简化代码，而且有利于程序调试，增强程序可读性。

功能的使用可分为无参数调用和有参数调用。

无参数调用就是 FC 块不从外部或者主调程序中接收参数，也不向外部发出参数。在 FC 中直接使用绝对地址完成控制程序的编程，如图 7-83 网络 1 所示 FC1 的应用。这种方式一般应用于分块结构的程序编写，每个功能实现整个控制任务的一部分，不重复调用。

有参数调用则需要从主调程序接收参数，将接收到的参数处理完毕后将处理结果再返还给调用程序，如图 7-83 网络 2 所示 FC2 的应用。有参数调用是在编辑 FC 功能块时在局部变量声明表内定义了形参，并使用虚拟的符号地址完成控制程序的编写，以便在其他块中能重复调用有参功能 FC。这种方式一般用于结构化程序编写。

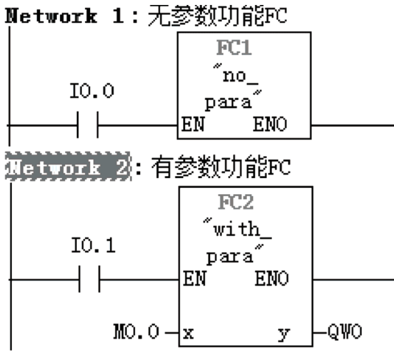


图 7-81　功能的使用

【例 7-22】 根据控制要求实现水塔高位水箱水位的自动控制。

水塔水箱水位自动控制系统如图 7-84 所示。其中 Lgh、Lgl 分别为水塔高位水箱的高、低水位传感器；Ldh、Ldl 为低位补给水箱的高、低水位传感器；YV 为电磁阀，用来控制补

水阀门；M 为电动机，用来拖动水泵对高位水箱供水。

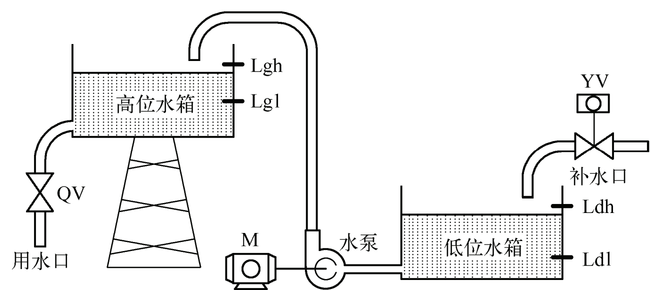


图 7-84　水塔高位水箱水位控制示意图

高位水箱的控制：为保证用户用水，要求始终保持高位水箱水位在 Lgl 与 Lgh 之间。当高位水箱水位低于 Lgl 时，水泵起动，从低位水箱抽水补给高位水箱，到达水位 Lgh 时，水泵自动停止。

低位水箱的控制：当水位低于低位供给水箱 Ldl 时，电磁阀 YV 得电后开启，对低位水箱补水，水位到达 Ldh 时，电磁阀 YV 失电，阀门闭合，停止补水。

1．控制要求

1）当低位补给水箱水位低于低水位时（Ldl 为 ON），电磁阀 YV 打开补水（YV 为 ON），定时器开始定时（设时为 2s），2s 以后，如果 Ldl 仍为 ON，则阀 YV 指示灯 XD 闪烁，表示阀 YV 没有进水，出现故障；如正常，则补水至高水位传感器 Ldh 为 ON 后，阀 YV 关闭（YV 为 OFF）。

2）当低水箱水位高于低水位（Ldl 为 OFF），且水塔水位低于低水位界时（Lgl 为 ON），电动机 M 运转，开始抽水补给高位水箱。当水塔水位高于水塔高水位界时（Lgh 为 ON），电动机 M 停止。

2．设计分析

根据控制要求有四个输入器件分别为水位传感器 Lgh、Lgl 和 Ldh、Ldl；三个输出负载分别为电磁阀 YV、电动机 M 和指示灯 XD，PLC 的 I/O 地址分配见表 7-17。

表 7-17　PLC 的 I/O 地址分配表

PLC 的 I/O 地址	连接的外部设备	在控制系统中的作用
I0.0	水位传感器 Lgh	高水箱高水位测量
I0.1	水位传感器 Lgl	高水箱低水位测量
I0.2	水位传感器 Ldh	低水箱高水位测量
I0.3	水位传感器 Ldl	低水箱低水位测量
Q0.0	电动机 M 主接触器线圈	水泵工作
Q0.1	电磁阀 YV	补水阀门工作
Q0.2	指示灯 XD	故障指示

整个控制程序根据功能可划分为 4 个部分，即闪烁控制功能（FC1）、低水箱水位控制功能（FC2）、高水箱水位控制功能（FC3）以及故障显示功能（FC4）。程序设计步骤如下。

（1）建立 FC 模块

在 SIMATIC Manager 界面右边显示栏，分别建立 FC1、FC2、FC3 及 FC4 功能，如图 7-85 所示。

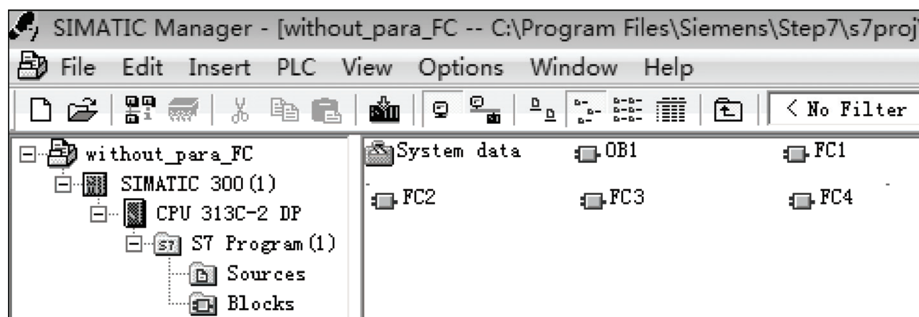


图 7-85　FC 模块的建立

（2）编写 FC 程序

双击图 7-85 所示 FC1 模块，进入 FC1 模块的程序编写界面，如图 7-86 所示，界面的上半部分为变量声明表，下半部分为程序指令部分。

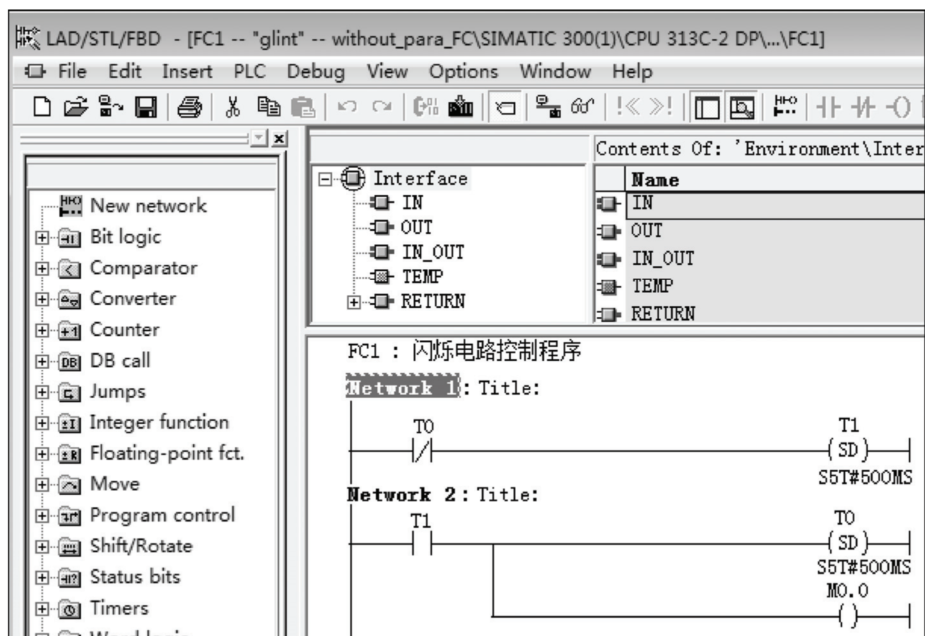


图 7-86　FC1 程序

通过网络 1 和网络 2 逻辑关系，M0.0 可以获得周期为 1s 的方波波形，可以用 M0.0 的动合触点来接通指示灯 Q0.2，使得 Q0.2 状态具有与 M0.0 同样的变化频率，用法如图 7-89 所示。

分别打开 FC2、FC3、FC4 功能模块，编写 FC2、FC3、FC4 程序如图 7-87~图 7-89 所示。

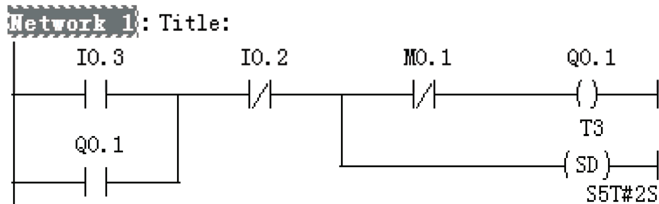


图 7-87　FC2 程序

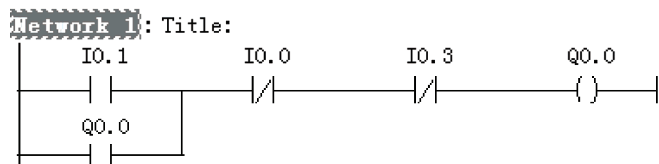


图 7-88　FC3 程序

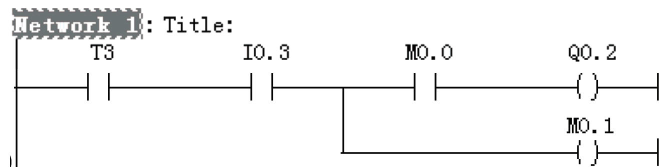


图 7-89　FC4 程序

在图 7-87 中，当低水箱液位低于低水位 I0.3 时，电磁阀打开给水箱充水。由于水位淹没低水位开关后，低水位开关 I0.3 要改变状态，不能再维持 Q0.1 接通，所以需要 Q0.1 实现自保持；当水位高于水箱的高水位时，高水位开关动作即 I0.2 得电，其动断触点断开，Q0.1 失电，电磁阀关闭，停止进水；若电磁阀故障，高水位开关 I0.2 不动作，则 2s 后 T3 的动合触点闭合，如图 7-89 所示，指示灯电路接通，同时辅助继电器 M0.1 线圈得电，则电磁阀也会关闭。

在图 7-88 中，当水塔高位水箱的水位低于低水位界 I0.1 时，电动机 M 工作，水泵从低水位往水塔抽水。由于水位淹没低水位开关后 I0.1 要改变状态，不能再维持 Q0.0 接通的状态，所以需要 Q0.0 实现自保持；当水位高于高水位界时高水位开关动作即 I0.0 得电，其动断触点断开，Q0.0 失电，电动机停止工作。若低水箱水位在低水位界之下（I0.3 为 ON），即使高位水箱需要抽水也不能起动电动机工作。

在图 7-89 中，T3 的动合触点闭合，说明电磁阀已经接通 2s，若低水箱低水位开关仍然没有改变状态，则电磁阀 YV 出现故障，故障指示灯 XD 亮。为满足指示灯闪烁要求，将时钟脉冲的触点 M0.0 也作为 Q0.2 动作的条件。

（3）OB1 程序

在 OB1 主程序中无条件调用功能 FC1~FC4，如图 7-90 所示，程序结构如图 7-91 所示。

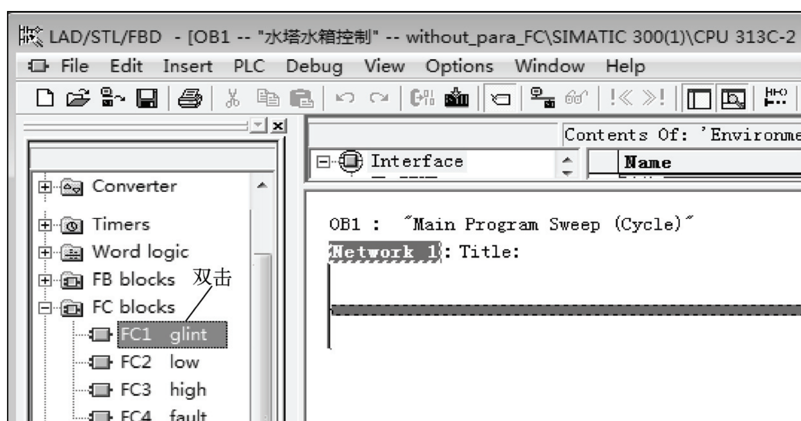


图 7-90　建立 OB1 程序

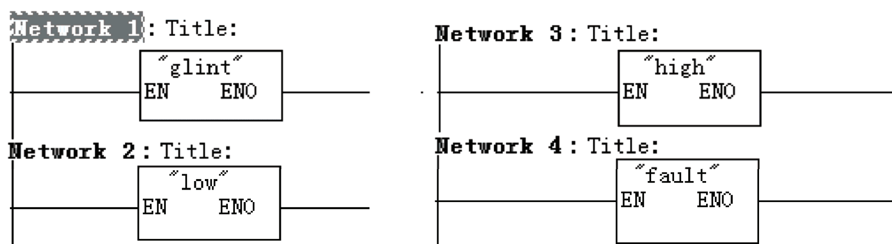


图 7-91　OB1 程序

（4）系统联调

将项目下载至 CPU 中，包括硬件配置、OB1 程序、FC1、FC2、FC3 以及 FC4 功能程序，在线调试和运行系统。

【例 7-23】 设计函数模块，计算 $Y = AX^2 + BX + C$ 值，且函数系数可变。

1）在 SIMATIC Manager 界面右侧显示栏，建立 FC1 模块，如图 7-92 所示。

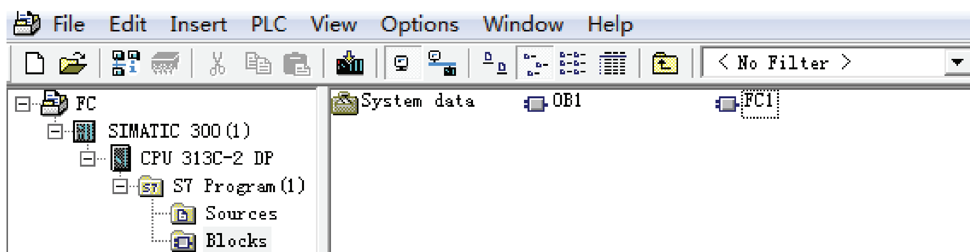


图 7-92　FC1 模块的建立

2）双击图 7-92 所示 FC1 模块，进入 FC1 模块的程序编写界面，如图 7-93 所示。

为 FC1 功能分别定义 Interface 参数，如图 7-94~图 7-96 所示。当设置输入、输出和输入/输出类型参数时，用户需要在程序声明中声明块调用的“接口”；当变量声明后，就会在本地数据堆栈中为临时变量保留一个有效的存储空间。

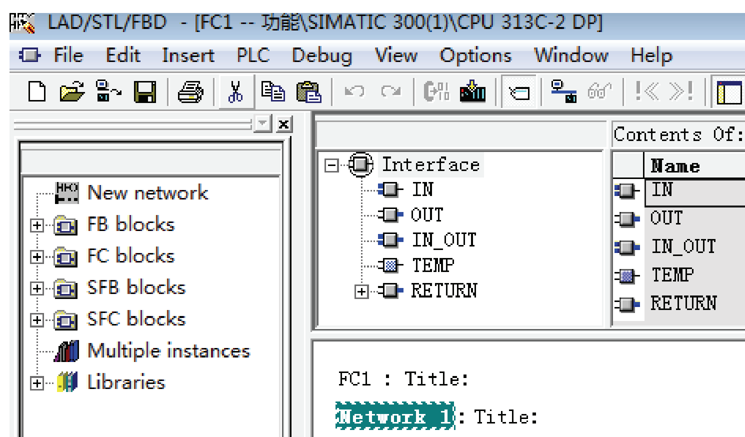


图 7-93　FC1 程序界面

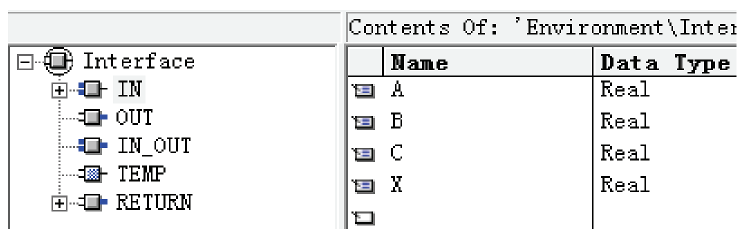


图 7-94　建立输入参数

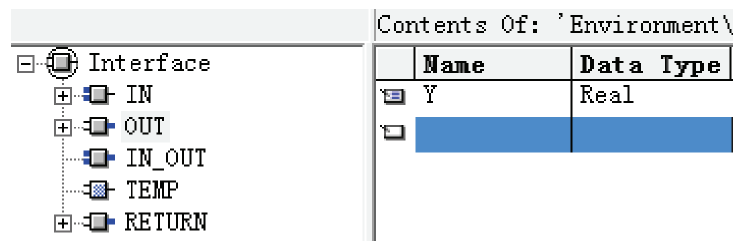


图 7-95　建立输出参数

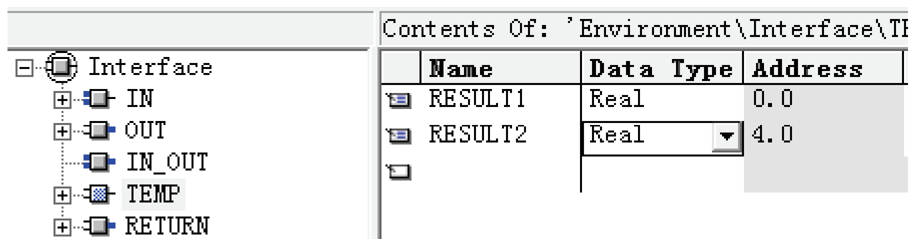


图 7-96　建立临时变量

3）编写 FC1 程序。为满足 $Y = AX^2 + BX + C$ 计算功能，编写 FC1 程序如图 7-97 所示。

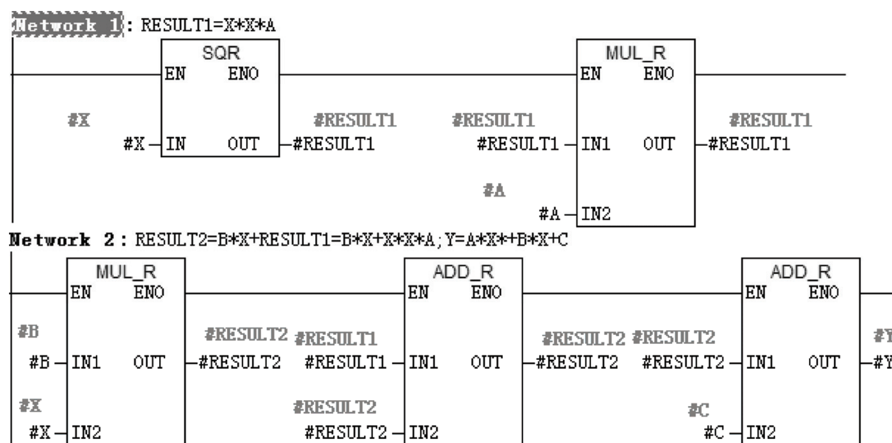


图 7-97　FC1 程序

4）OB1 程序。在 OB1 主程序中调用功能 FC1，如图 7-98 所示，程序在线运行结果如图 7-99 所示。

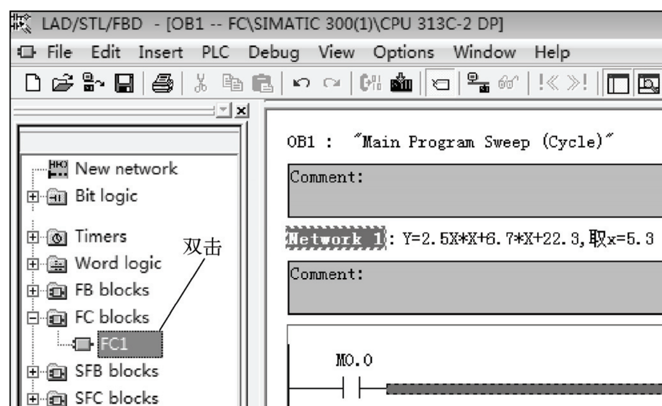


图 7-98　FC1 的调用

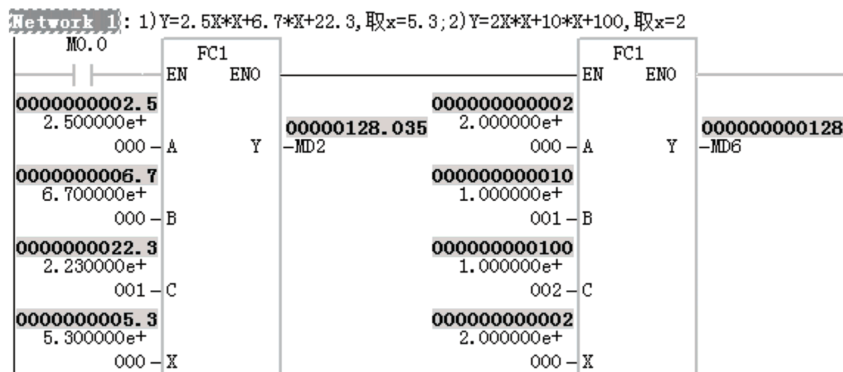


图 7-99　OB1 在线监控

由于 FC 功能块的逻辑计算完全靠模块的输入/输出接口上的地址提供数据源，若想修改 A、B、C 的值，必须也把 A、B、C 作为模块的输入变量。

7.9.3　FB 及其应用

FB 是功能块，拥有自己的存储区，即背景数据块。

FB 在程序的体系结构中位于组织块之下，包含程序的一部分，这部分程序可以在 OB1 中多次被调用。功能块的所有参数和静态参数都存储在一个单独的、被指定给该功能块的数据块中，即背景数据块。当调用 FB 时，该背景数据块会自动打开；当块退出时，背景数据块中的数据仍然保持。

每次调用 FB 都需要一个 DB 来供其存放逻辑运算数据，而不像 FC 需要输入/输出接口地址作为数据源。

【例 7-24】 设计函数模块，计算 $Y = AX^2 + BX + C$ 值，且函数系数可变。

1）在 SIMATIC Manager 界面右侧显示栏，建立 FB1 模块，如图 7-100 所示。

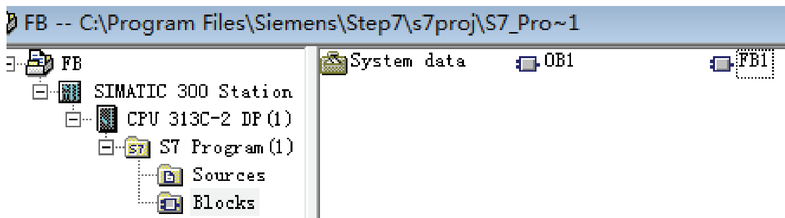


图 7-100　FB1 模块的建立

2）双击图 7-100 所示 FB1 模块，进入 FB1 模块的程序编写界面，如图 7-101 所示，为 FB1 功能块分别定义 Interface 参数，如图 7-102 所示。

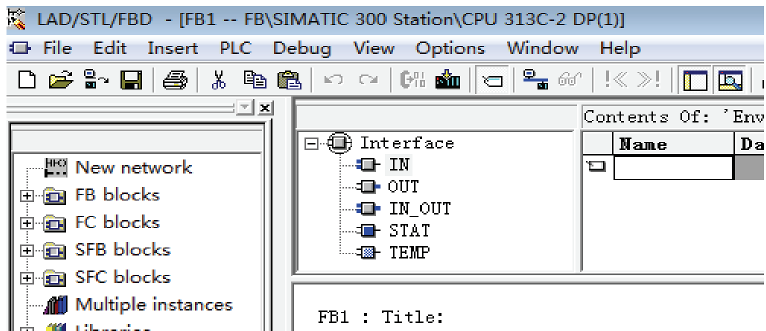
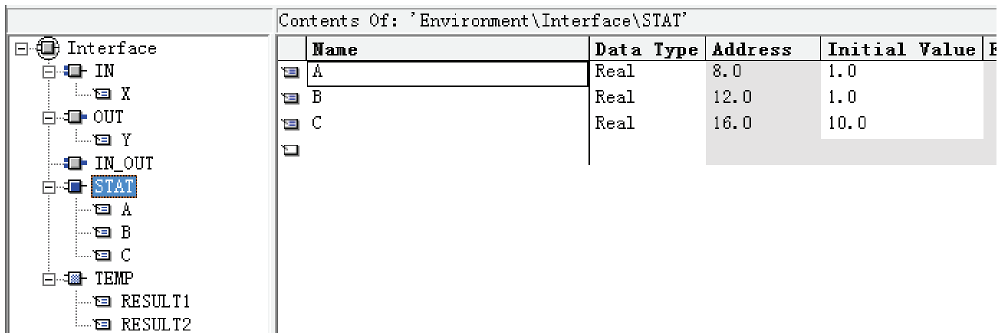


图 7-101　FB1 程序界面



Name	Data Type	Address	Initial Value
A	Real	8.0	1.0
B	Real	12.0	1.0
C	Real	16.0	10.0

图 7-102　建立接口参数

从图 7-102 可见，建立输入参数 X；输出参数 Y；静态变量 A、B、C，并设其初值分别为 A=1、B=1、C=10；临时变量 RESULT1、RESULT2。变量声明后，除了在本地数据堆栈中为临时变量保留一个有效存储空间外，还要为静态变量保留空间。

3）FB1 程序编写。为满足 $Y = AX^2 + BX + C$ 功能，编写 FB1 逻辑关系如图 7-103 所示。

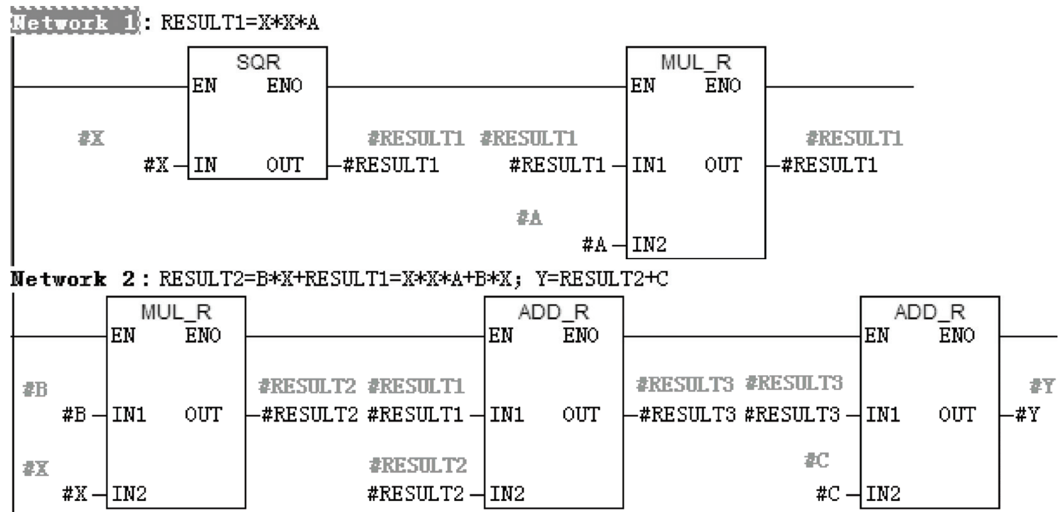


图 7-103　FB1 程序

4）为 FB1 分别建立背景数据块 DB1、DB2。在 SIMATIC Manager 界面右侧显示栏，建立 DB1 模块，弹出界面如图 7-104 所示，在下拉菜单中选择“Instance DB”选项，采用同样的路径和方法建立 DB2。

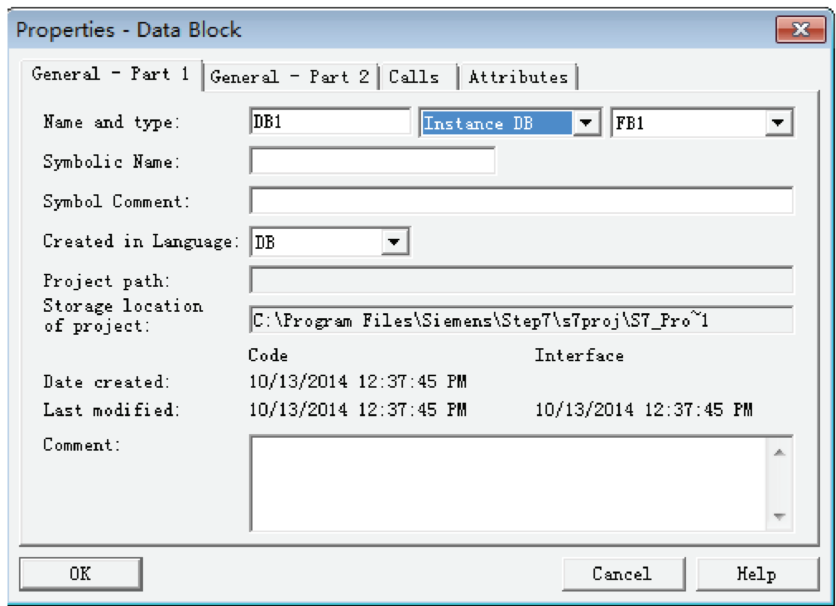


图 7-104　FB1 背景数据块 DB1 的建立

也可以在 OB1 中调用 FB1 模块时自动生成背景数据块，操作步骤如图 7-105 所示。



图 7-105 调用 FB 时自动生成背景数据块

5）OB1 程序的编写。在 OB1 主程序中调用功能 FB1，如图 7-106 所示，程序编写如图 7-107 所示。

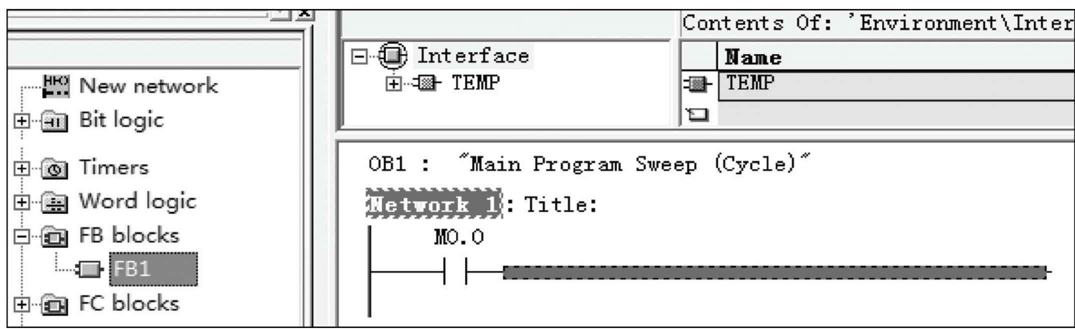


图 7-106 FB1 的调用

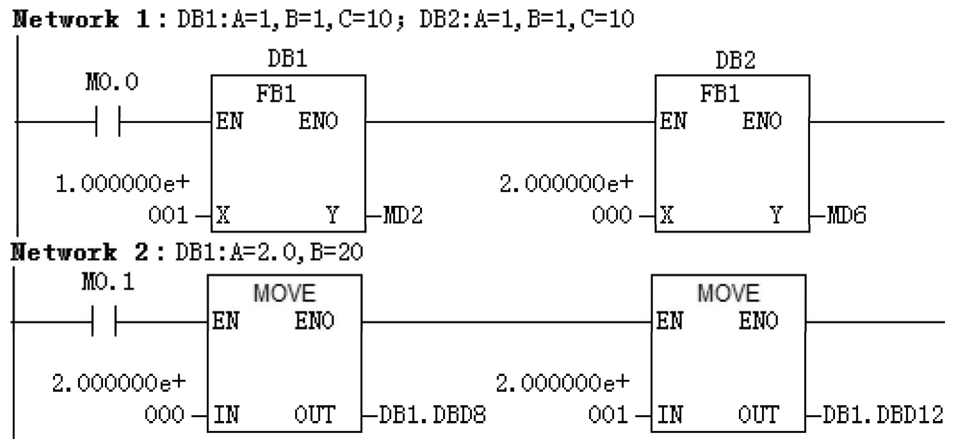


图 7-107 OB1 程序

网络 1 程序在线运行数据如图 7-108 所示，当 M0.0 = 1 时，$MD2 = X^2 + X + 10 = 10*10 + 10 + 10 = 120$，$MD6 = 2^2 + 2 + 10 = 16$，DB1 数据表在线监控情况如图 7-109 所示。

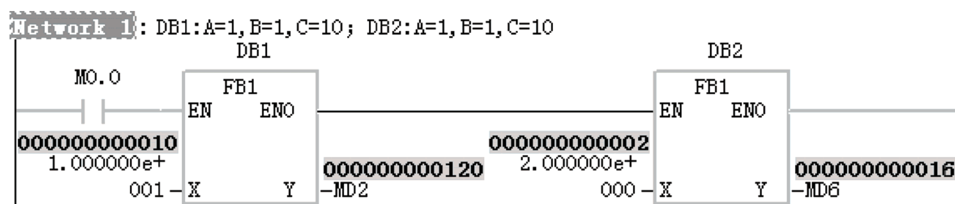


图 7-108　OB1 中 FB1 模块运行

@DB1 -- FB\SIMATIC 300 Station\CPU 313C-2 DP(1) ONLINE

	Addre	Declaration	Name	Type	Initial valu	@Actual valu	Actual value
1	0.0	in	X	REAL	0.000000e+000	10.0	0.000000e+000
2	4.0	out	Y	REAL	0.000000e+000	120.0	0.000000e+000
3	8.0	stat	A	REAL	1.000000e+000	1.0	1.000000e+000
4	12.0	stat	B	REAL	1.000000e+000	1.0	1.000000e+000
5	16.0	stat	C	REAL	1.000000e+001	10.0	1.000000e+001

图 7-109　DB1 数据表在线监控值

从图 7-109 中还可以看出，除了临时变量不出现在背景数据块中，其他声明的变量都要在背景数据块中的数据结构中声明。对于初值（Initial Value），用户可以在功能块的变量声明中输入；如果没有输入，则软件给出默认值；当数据块第一次存盘时若用户没有明确地声明实际值，则初值将被用于实际值。

如图 7-110 所示，OB1 的网络 2 为改变 DB1 数据块中的静态变量值。当 M0.1=1 时，将实数 2.0 传送给 DB1.DBD8，从图 7-109 可见，DB1.DBD8 是静态变量 A 的地址；同理，将实数 20.0 传送给静态变量 B，其在 DB1 数据块中的地址是 DB1.DBD12。观察网络 1 的 DB1 背景数据块所在 FB1 的在线监控值，可见 Y=410.0，与实际计算公式 $MD2 = 2*X^2 + 20*X + C = 2*10^2 + 20*10 + 10 = 410$ 计算结果相一致。

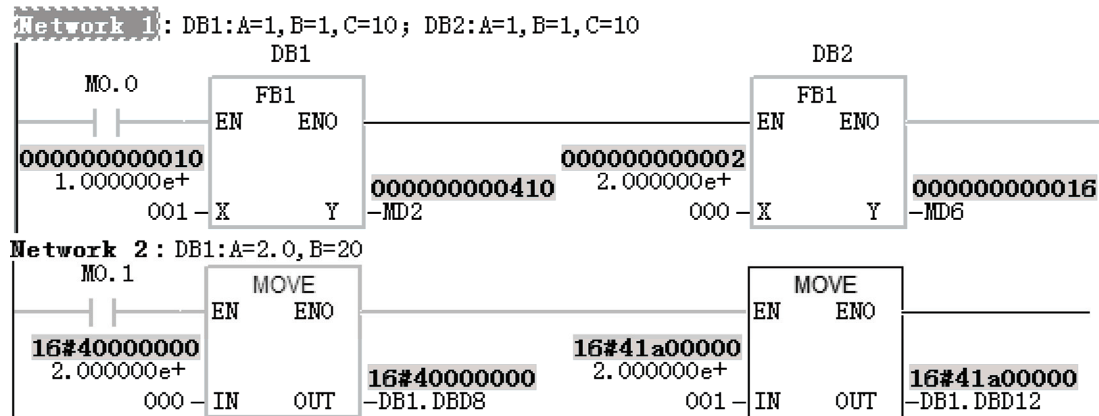


图 7-110　修改系数后 OB1 中 FB1 模块运行

由于 DB2 数据块静态变量并未改变，因此其功能块运行结果 $MD6 = Y = X^2 + X + 10 = 2^2 + 2 + 10 = 16$，与在线监控结果保持一致。

DB1 数据块变量修改后的运行状态如图 7-111 所示，当系统停止运行时，数据块中保持所有数据状态。

	Addre	Declaration	Name	Type	Initial valu	@Actual value	Actual value
1	0.0	in	X	REAL	0.000000e+000	10.0	0.000000e+000
2	4.0	out	Y	REAL	0.000000e+000	410.0	0.000000e+000
3	8.0	stat	A	REAL	1.000000e+000	2.0	1.000000e+000
4	12.0	stat	B	REAL	1.000000e+000	20.0	1.000000e+000
5	16.0	stat	C	REAL	1.000000e+001	10.0	1.000000e+001

图 7-111　修改后的 DB1 数据监控

第 8 章　S7-300 PLC 程序设计及应用

8.1　程序结构

用户程序结构可分为线性程序、分块程序和结构化程序 3 种。

1．线性程序

所谓线性程序结构，就是将整个用户程序连续放置在一个循环程序块（OB1）中，块中的程序按顺序执行，CPU 通过反复执行 OB1 来实现自动化控制任务。这种结构与 PLC 所替代的继电器-接触器控制类似，CPU 逐条地处理指令。

所有的程序都可以用线性结构实现，不过，线性结构一般用于相对简单的程序编写。对于一些控制规模较大，运行过程比较复杂的控制程序，特别是分支较多的控制程序不宜选择这种结构。

2．分块程序

分块程序是根据工程特点，把一个复杂的控制工程分成多个比较简单的、规模较小的控制任务，每一个控制任务分配给一个程序块，在程序块中编制具体任务的控制程序，最后由主程序 OB1 按顺序调用各个功能块，并控制程序的执行，如图 8-1 所示。

在分块程序中，既无数据交换，也不存在重复利用的程序代码。功能（FC）和功能块（FB）不传递也不接收参数，分块程序结构的编程效率比线性程序有所提高，程序测试也比较方便，对程序员的要求也不太高。对不复杂的控制程序可考虑采用这种程序结构。

3．结构化程序

所谓结构化程序（或模块化编程），就是处理复杂自动化控制任务的过程中，为了使任务更易于控制，常把过程要求类似或相关的功能进行分类，分割为可用于几个任务的通用解决方案的小任务，这些小任务可以用相同的程序段来表示，不同的参数可以配有不同的数据块来区分和替代。OB1 通过调用这些程序块来完成整个自动化控制任务。如图 8-2 所示。

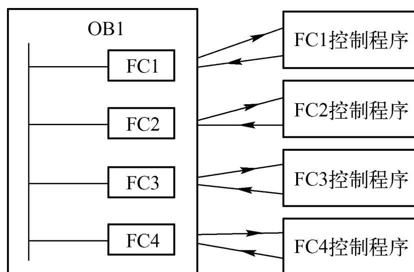


图 8-1　分块程序结构

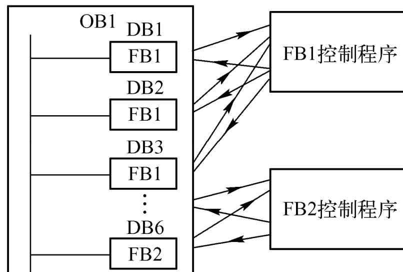


图 8-2　结构化程序结构

结构化程序的特点是每个块（FC 或 FB）在 OB1 中可能会被多次调用，以完成具有相同过程工艺要求的不同控制对象。这种结构可简化程序设计过程、减小代码长度、提高编程效率，比较适合于较复杂的自动化控制任务设计。

8.2 具有多种工作方式的机械手 PLC 控制系统的设计

机械手是一种能模拟人的手臂的部分动作，按预定的程序、轨迹及其他要求，实现抓取、搬运工件或操纵工具的自动化装置。它可代替人的繁重劳动以实现生产的机械化和自动化，能在有害环境下工作，以保护人身安全。

1．控制要求

如图 8-3 所示，在水平、垂直位移的机械设备中，用机械手将工件从 A 工作台搬至 B 工作台，机械手的动作采用气缸驱动，气缸的动作由电磁阀进行换向控制。

从原点（指机械手在最上面、最左面、手爪松开时）开始，机械手经过下降、夹紧、上升、右移、下降、松开、上升、左移 8 个动作完成一个循环并返回到原点（初始位置）从而实现一次搬运功能，具体动作步骤如下：

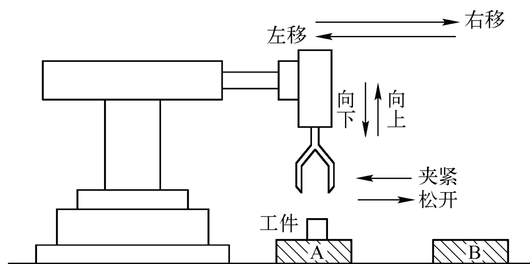


图 8-3　搬运机械手的动作示意图

1）按下启动按钮，系统开始工作。

2）机械手首先向下运动，运动到最低位置停止（由下限位开关确定）。

3）机械手开始夹紧工件，一直到把工件夹紧为止（由接近开关确定）。

4）机械手开始向上运动，一直运动到最上端（由上限位开关确定）。

5）上限位开关闭合后，机械手开始向右运动。

6）运行到右端（右限位开关）后，机械手开始向下运动。

7）向下运行到下限位开关后，机械手把工件松开（由接近开关确定）。

8）工件松开后，机械手开始向上移动，直到触动上限位开关。

9）机械手开始向左运动，直到触动左限位开关，此时机械手回到初始位置。

按照机械手的操作流程，实际控制时，可将系统设为手动、回原点和自动 3 种工作方式。

1）手动方式：系统可以通过外部操作按钮完成每个动作的独立操作。

2）回原点方式：系统在进行自动控制操作前，必须保证机械手处于原点位置，该方式可以使机械手自动返回原点，为自动操作做好准备。

3）自动工作方式：该方式下又设置了单步、单周期和连续运行的工作方式。

① 单步方式，指系统每次按下启动按钮，机械手只运动一个工序，主要用于机械手的调试检修。

② 单周期方式，指系统每次按下启动按钮，机械手完成从下降、抓物、上升、右移、下降、放物、上升、左移回原点的一个完整周期后停止。

③ 连续运行方式，指系统启动后，机械手完成一个完整运行周期后，继续不停歇地进行下一个周期，周而复始，直至按下停止按钮。

这三种方式都有一个共同点，即严格按照机械手的工作步序行进，只有当前步序完成且转换条件满足才能进入下一个工序。

2．PLC 的选型

按照控制要求，本系统设有手动、回原点、单步、单周期和连续运行共 5 种工作方式，

且各种方式不能并存，故需使用选择开关来选择工作方式，占 PLC 的 5 个输入点。

手动工作方式是指利用按钮对机械手每一步动作单独进行控制，需要控制上升、下降、左行、右行、夹紧、松开六个动作的按钮，占用 PLC 的 6 个输入点。

回原点工作方式是系统工作前，实现机械手自动回到初始状态的操作，即让机械手到达最上面、最左面且手爪松开的状态（原点），为后续的自动工作方式做准备。需要设置一个专门的回原点启动按钮，占 PLC 的 1 个输入点。

自动工作方式包含单步、单周期和连续 3 种方式，需要设置启动和停止两个按钮，占 PLC 的 2 个输入点。其中：

单步工作方式：从原点开始，按自动工作循环的工序，每按下一次启动按钮，机械手就完成一步动作；如果没有按下启动按钮，则系统不会转换到下一步。

单周期工作方式：从原点开始，按下启动按钮后，机械手按工序自动完成一个周期的动作后，停在原点。即按一次启动按钮，系统工作一个周期，返停在原点。

连续工作方式：按下启动按钮后，机械手从原点开始一个周期接一个周期地反复连续工作。在工作中若按下停止按钮，则机械手不会立即停止工作，在完成当前工作周期的全部操作后，系统返回并停留在原点。

此外机械手动作转换需要 4 个限位开关和 2 个接近开关，这 6 个开关也要占 PLC 的 6 个输入点。所以 PLC 控制系统的输入信号有 20 个，均为开关量。

本设计中 PLC 控制系统的输出为 6 个开关量信号，其中：

垂直、水平移动采用双控气阀，包括下降电磁阀、上升电磁阀、左移电磁阀、右移电磁阀；手爪采用单控气阀，即夹紧/松开电磁阀，电磁阀得电时，抓紧工件，失电时，手爪松开；为了对机械手原点位置进行指示，系统还配置了一个原点指示灯。

根据以上分析所确定的用户输入设备及输出设备可知，PLC 共需要 20 点输入、6 点输出。

PLC 可以选择紧凑型的 CPU，该类 PLC 自带有数字量和模拟量的输入输出，也可以通过选择 CPU 模块并在机架上添加数字量输入/输出模块等方案来解决，配置方法灵活多样。本系统选择紧凑型的 CPU314C-2DP 型号，该模块集成了 24 路数字量输入和 16 路数字量输出，可以满足控制要求，同时留有一定的裕量。

3．I/O 地址分配

将 20 个输入信号、6 个输出信号按各自的功能类型进行分配，并与 PLC 的 I/O 点一一对应，编排地址。表 8-1 是外部输入/输出设备与 PLC 的 I/O 点的地址编号对照表。

表 8-1　PLC 的 I/O 地址分配表

PLC 的 I/O 地址	连接的外部设备	在控制系统中的作用
I0.0	SQ1	下限位开关
I0.1	SQ2	上限位开关
I0.2	SQ3	右限位开关
I0.3	SQ4	左限位开关
I0.4	SQ5	加紧
I0.5	SQ6	松开

（续）

PLC 的 I/O 地址	连接的外部设备	在控制系统中的作用	
I1.0	选择开关 SA	工作方式	手动运行
I1.1			回原点
I1.2			单步运行
I1.3			单周期运行
I1.4			连续运行
I1.5	SB0	回原点起动	
I1.6	SB1	自动	启动按钮
I1.7	SB2		停止按钮
I2.0	SB10	手动	下降
I2.1	SB11		上升
I2.2	SB12		右移
I2.3	SB13		左移
I2.4	SB14		夹紧
I2.5	SB15		松开
Q0.0	YV0	下降动作控制	
Q0.1	YV1	上升动作控制	
Q0.2	YV2	右移动作控制	
Q0.3	YV3	左移动作控制	
Q0.4	YV4	夹紧/松开动作控制	
Q0.5	HL	原点指示	

4．程序设计

（1）程序的总体结构设计

将系统的程序按工作方式和所完成的功能不同可分成四部分，包括公共程序、手动程序、回原点程序和自动程序。为了便于程序编写及维护管理，采用 FC 功能编写程序，用工作方式的选择信号作为 OB1 中 FC 执行的条件；增加热启动模块 OB100，以便初始化自动程序。采用启保停电路的编程方法编制程序，程序结构如图 8-4 所示。

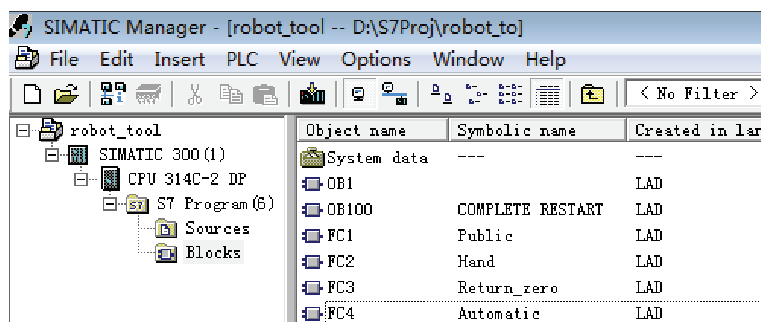


图 8-4　程序结构

（2）内部资源分配

编程时，对需要使用的辅助继电器 M 和时间继电器 T 进行分配，有助于提高编程效率，并减少资源冲突。具体分配见表 8-2。

表 8-2　PLC 内部资源分配

名　称	作　用	名　称	作　用
M0.0	原点状态标志位	M1.0	（自动）左侧下移
M0.1	连续运行状态标志位	M1.1	（自动）夹紧工件
M0.2	允许工序转换标志位	M1.2	（自动）左侧上移
M0.3	自动模式初始状态	M1.3	（自动）右移
M2.0	（回原点）上移标志位	M1.4	（自动）右侧下移
M2.1	（回原点）左移标志位	M1.5	（自动）松开工件
M2.5	系统初始化	M1.6	（自动）右侧上移
		M1.7	（自动）左移（回原点）

（3）OB1 程序

主程序中需要调用功能 FC1/FC2/FC3/FC4，调用条件如图 8-5 所示。

FC1 为公共程序，主要用于初始状态的设置和工作方式功能程序的跳转；FC2 为手动控制子程序，当 SA 置手动档，I1.0 接通，执行该功能程序；FC3 为回原点控制子程序，当 SA 置回原点档，I1.1 接通，执行该功能程序，可将机械手拉回原点；FC4 为自动控制功能程序，当 SA 置单步、单周期、连续三种中的任一档位，I1.2 或 I1.3 或 I1.4 接通，执行该程序。

（4）OB100 程序

OB100 为热启动程序，启动程序如图 8-6 所示，每次热启动时执行该程序，使得 M2.5=1，为公共程序提供一个初始化信号，功能见公共程序网络 2。

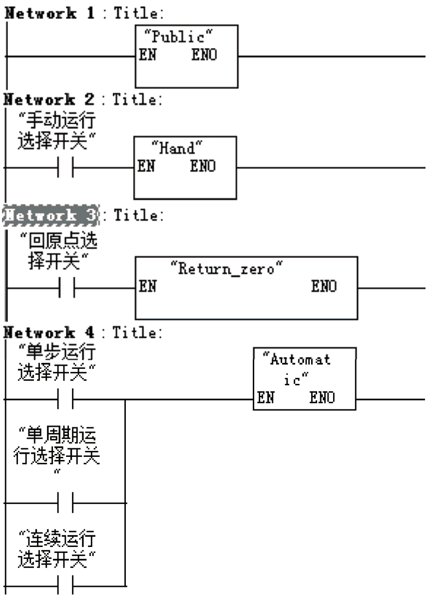


图 8-5　OB1 程序

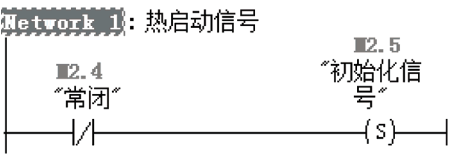


图 8-6　OB100 程序

（5）公共程序 FC1

图 8-7 为机械手控制公共程序，主要包括：设置原点条件 M0.0 及驱动原点指示灯 Q0.5；在开机运行时先执行 OB100 程序产生一个初始化脉冲，或手动（I1.0=1）、回原点（I1.1=1）状态时，将 M0.3 置位，为系统进入自动方式做准备；复位自动程序中各步序的辅助继电器(MB1 = 0)；满足相应条件时，调用相关功能程序。

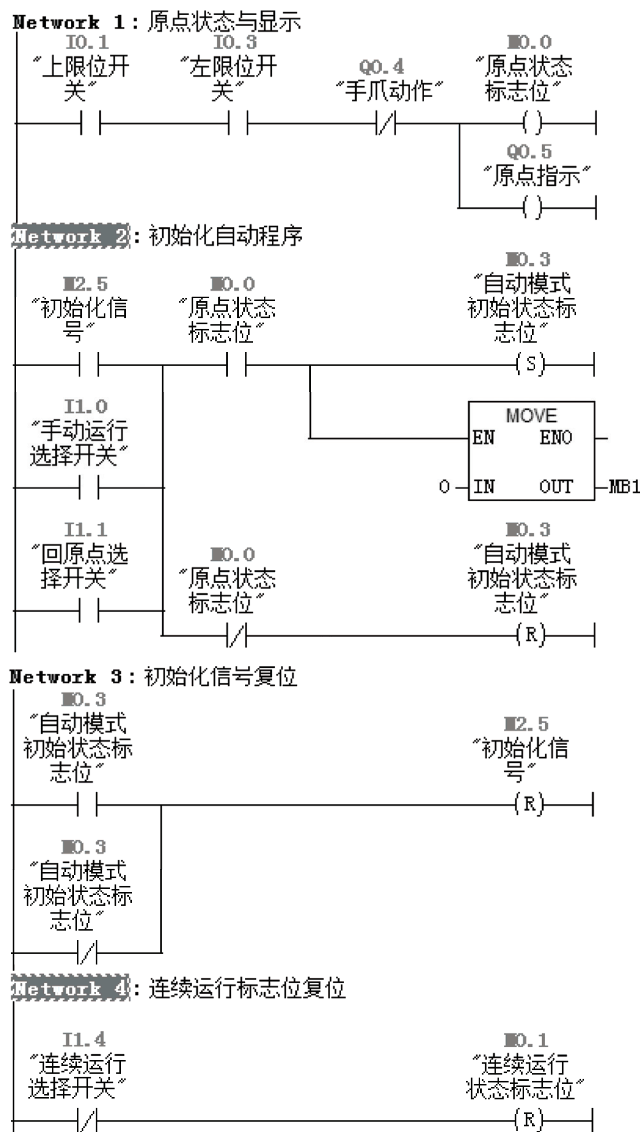


图 8-7　机械手公共程序

（6）手动程序 FC2

实现手动功能的程序相对较为简单，一般采用经验设计法进行设计，手动控制功能的实现如图 8-8 所示的子程序。手动按钮 I2.0~I2.5 分别控制下降、上升、右移、左移、夹紧和松开各个动作。由于机械手只有处于上限位置（I0.1=1）时，才允许左右移动，所以在左右移动

的控制环节中加入了 I0.1 作为上限联锁；同理在夹紧和松开动作环节中加入了下限位置 I0.0 作为联锁保护。

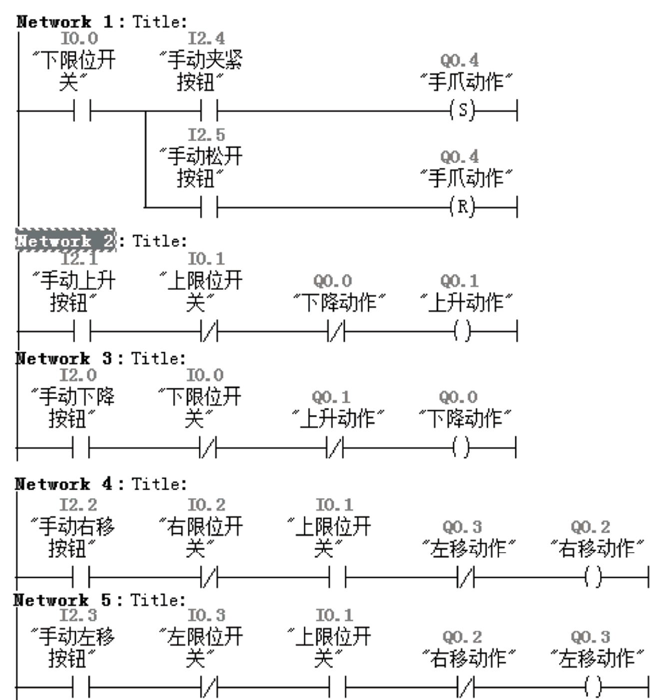


图 8-8　手动程序 FC2

（7）回原点程序 FC3

当系统启动时或因意外停止恢复操作时，机械手可能不在原点，这时可通过 FC3 功能程序使机械手自动返回原点。图 8-9 为自动回原点程序的顺序功能图，图 8-10 为对应的梯形图程序。

当 SA 选择回原点模式，按下回原点启动按钮（I1.5 为 ON），M2.0 为 ON，Q0.1 得电，机械手上升，到上限位开关 I0.1 动作时停止；此时，M2.1 为 ON，Q0.3 得电，机械手左移，到左限位开关 I0.3 动作后停止；并在左侧时，复位 Q0.4，手爪松开。这时，机械手原点条件满足，M0.0 为 ON，并将自动程序的初始步 M0.3 置位；为下续的自动程序（单步、单周期和连续模式）做好准备。

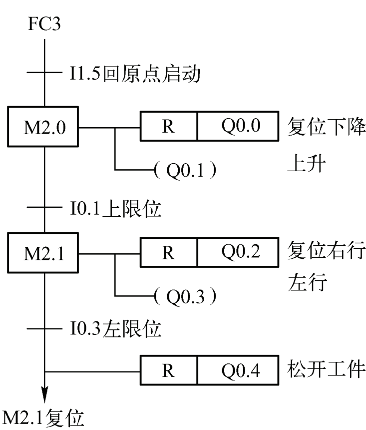


图 8-9　自动回原点程序的顺序功能图

（8）自动程序 FC4

自动程序相对比较复杂，对于顺序控制系统一般采用顺序控制设计法。

如图 8-11 所示，根据机械手的运动过程及控制要求，分别用辅助继电器 M0.3 及 M1.0~M1.7 表示各个工步，绘制出机械手运动控制的自动工作状态流程图，以表明动作的顺

序和转换条件，再根据工作状态流程图就可以很容易地编写出相应的梯形图。

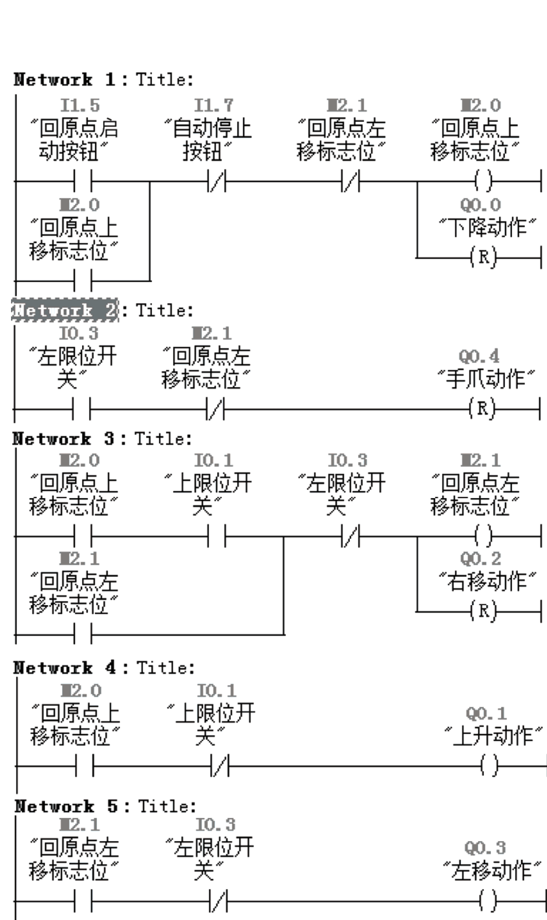


图 8-10　回原点程序 FC3

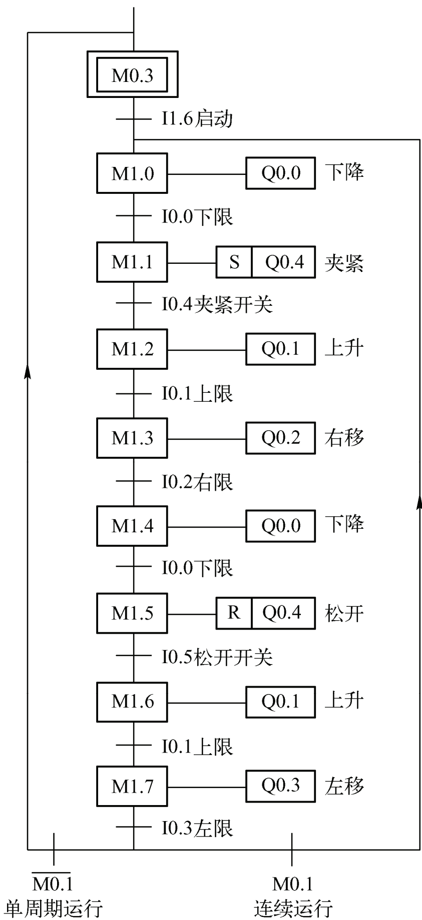


图 8-11　机械手自动控制流程图

对于顺序控制可用多种方法进行编程，例如置位/复位指令编程、启保停编程等方法，在这个控制系统中采用启保停编程。根据流程图可以看出，工步和工步之间的转移通过限位开关或电磁接近开关来实现，自动工作方式梯形图程序如图 8-12 所示，输出电路程序如图 8-13 所示。

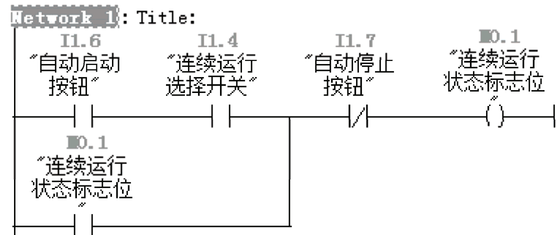


图 8-12　自动工作方式梯形图程序

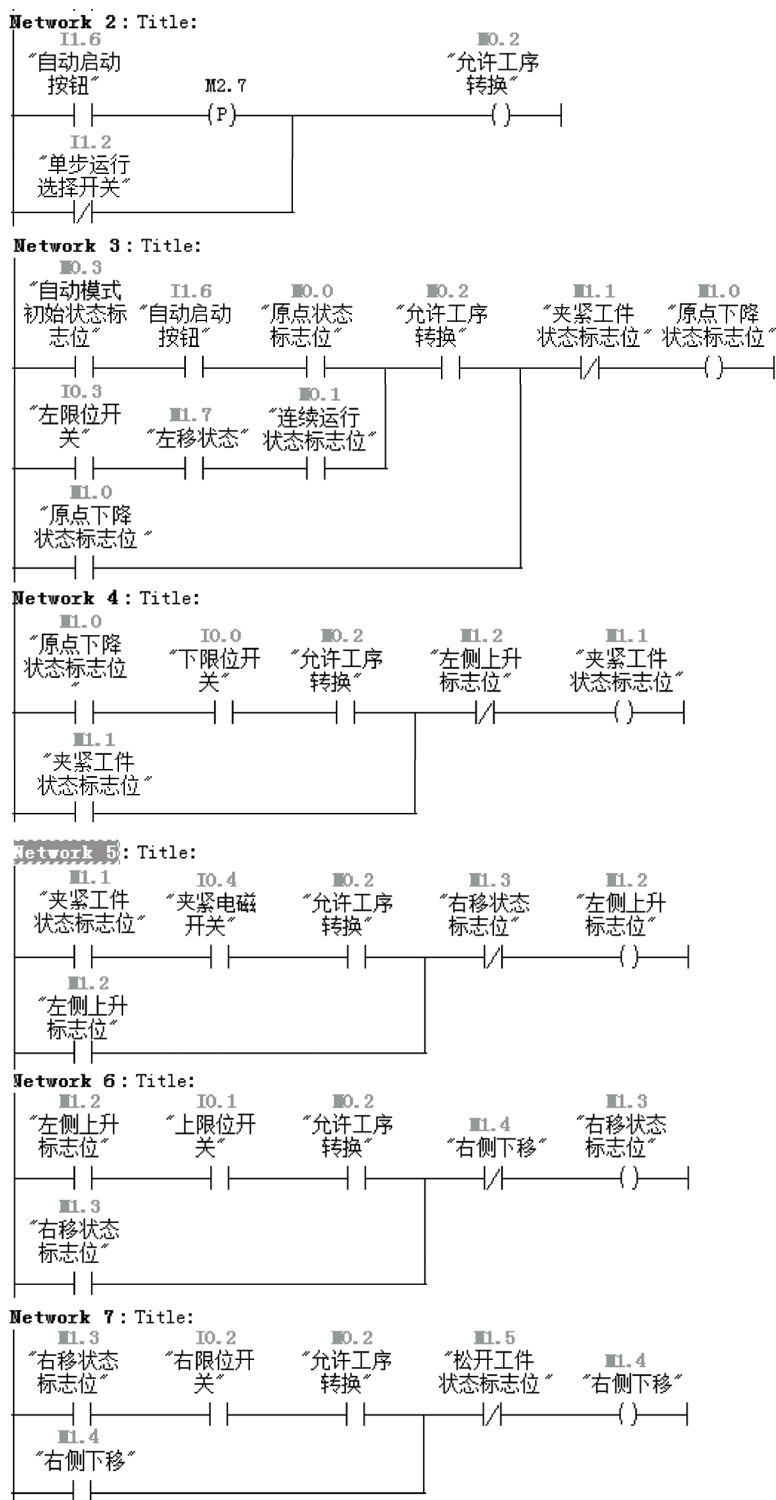


图 8-12　自动工作方式梯形图程序（续）

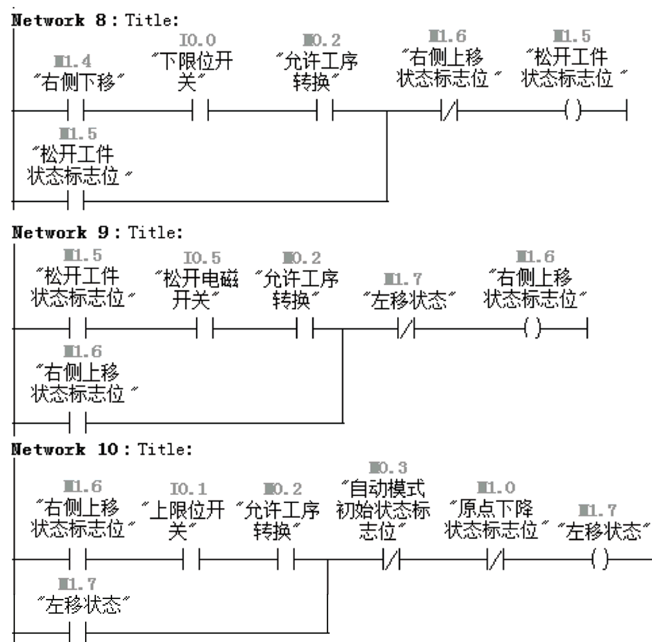


图 8-12　自动工作方式梯形图程序（续）

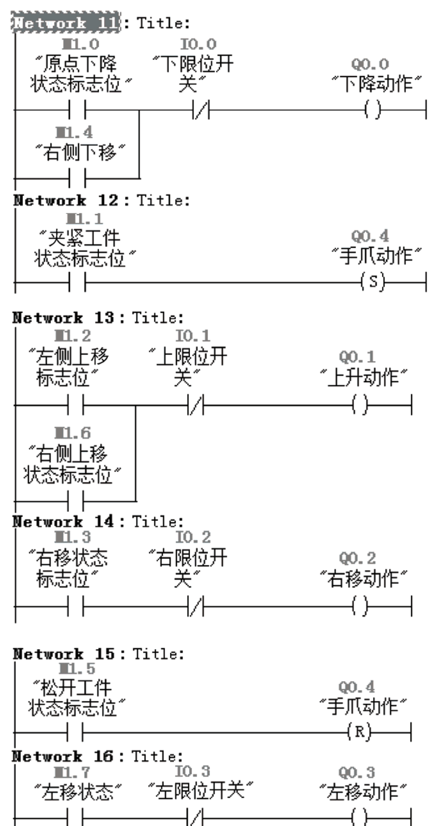


图 8-13　输出电路程序

机械手的自动程序中包括单步、单周期和连续工作三种工作方式，采用启保停进行设计时需要加以区分；这里采用了两个标志位来区分这三种工作方式，M0.1 为连续工作标志，M0.2 为允许步序转换标志。

观察图 8-12 所示的程序，单步运行方式时，连续工作标志位 M0.1 为 OFF，允许步序转换标志 M0.2 在启动按钮 I1.6 每次按下时，导通一个扫描周期；单周期运行方式时，连续工作标志 M0.1 为 OFF，允许步序转换标志 M0.2 为 ON；连续运行方式时，连续工作标志位 M0.1 为 ON，允许步序转换标志 M0.2 为 ON。

1）单步运行

当 SA 选择单步运行时，即 I1.2 为 ON，这种模式时连续工作标志 M0.1 为 OFF，允许步序转换标志 M0.2 只在启动按钮 I1.6 每次按下时，导通一个扫描周期。

所以当系统进入自动运行状态，即原点标志位 M0.0、初始状态 M0.3 均为 ON，按下启动按钮（I1.6 为 ON）后，则 M0.2 导通一个扫描周期，M1.0 为 ON；机械手下降到下限位开关（I0.0）动作后停止；因为 M0.2 仅导通了一个扫描周期，所以步序不能自动转换到 M1.1，只有再次按下启动按钮、M0.2 为 ON 后，M1.1 方能为 ON。下续各步序的转换同理，即每按压一次启动按钮（I1.6 为 ON），系统方能运行一个步序。

2）单周期运行

当 SA 选择单周期运行时，即 I1.3 为 ON，这时连续工作标志 M0.1 为 OFF，允许步序转换标志 M0.2 保持为 ON（因 I1.2 为 OFF）。

按照梯形图，系统在自动运行状态时，原点标志位 M0.0、初始状态 M0.3 均为 ON；这时按下启动按钮（I1.6 为 ON）后，M1.0 立刻为 ON；驱动机械手下降（Q0.0=1）到下限位（I0.0=1）；因为 M1.0、M0.2、I0.0 均导通，步序自动转换到 M1.1，机械手执行工件夹紧动作直到夹紧电磁开关 I0.4 动作；此时因 M1.1、M0.2、I0.4 均导通，步序自动转换到 M1.2，机械手上升（Q0.1=1）到上限位（I0.1=1）；以下各步的转换都是自动进行的。直到 M1.7，因为连续工作标志 M0.1 为 OFF，只能转换到 M0.3 步后停止。系统完成一个运行周期。

3）连续运行

当 SA 选择连续运行时，即 I1.4 为 ON，这时连续工作标志 M0.1 将在启动（I1.6 为 ON）后持续为 ON，允许步序转换标志 M0.2 也保持为 ON（因 I1.2 为 OFF）。

启动按钮按下（I1.6 为 ON）后，第一个周期各步的运行和转换与单周期运行方式相同；只是在最后一步 M1.7 时，因为连续工作标志 M0.1 为 ON，系统将直接转换到 M1.0 步，继续下一个周期的运行。直到按下停止按钮（I1.7 = 1）后，系统执行完一个完整周期的操作后停回原点。

4）输出电路

图 8-13 所示为自动控制程序的输出电路；通过各步序对应的辅助继电器 M1.0~M1.7 驱动输出继电器 Q0.0~Q0.4，实现机械手的控制功能。通过采用辅助继电器来统一驱动输出继电器，可有效避免双线圈的出现，使程序结构更为清晰、更易阅读。

该机械手在改用 PLC 控制以前，采用传统的继电器-接触器系统控制，其控制线路复杂，继电器多；采用 PLC 程序控制之后，控制线路简单化，省去了全部继电器，安

装十分方便，并且保证了系统运行的可靠性，减少了维修量，提高了工效，经济效益得到明显提高。

8.3 多台设备报警控制系统的设计与实现

8.3.1 控制要求

某生产单元有 4 台加工机床，每台机床有缺料呼叫按钮、CNC 求援呼叫按钮和报警复位按钮操作台，所有报警信号提供给车间广播系统。编写程序满足广播、立体仓库物料出库及调度车间物料小车等功能。

8.3.2 硬件配置及系统资源的分配

考虑生产现场已有多个生产单元，每个生产单元采用 S7-300 PLC 控制器且单元之间通过工业以太网通信，为了保持产品的一致性和联网方便，选择 CPU315-2PN/DP PLC。硬件配置如图 8-14 所示。

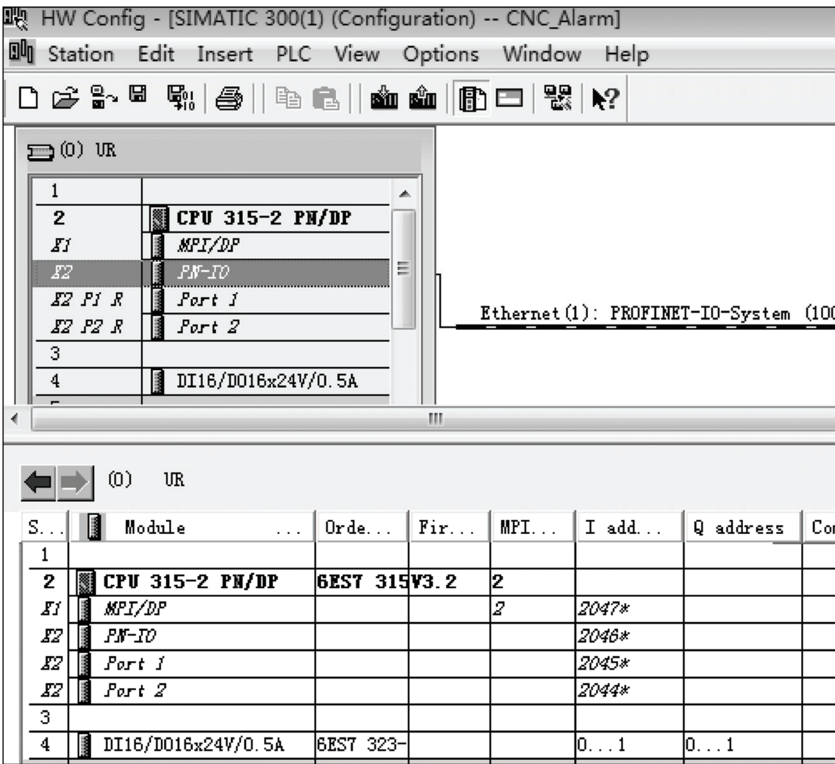


图 8-14 硬件配置界面

双击图 8-14 中“PN-IO”，弹出以太网属性设置界面，进行模块 IP 地址和子网掩码设置，完成界面如图 8-15 所示。

系统资源分配见表 8-3。

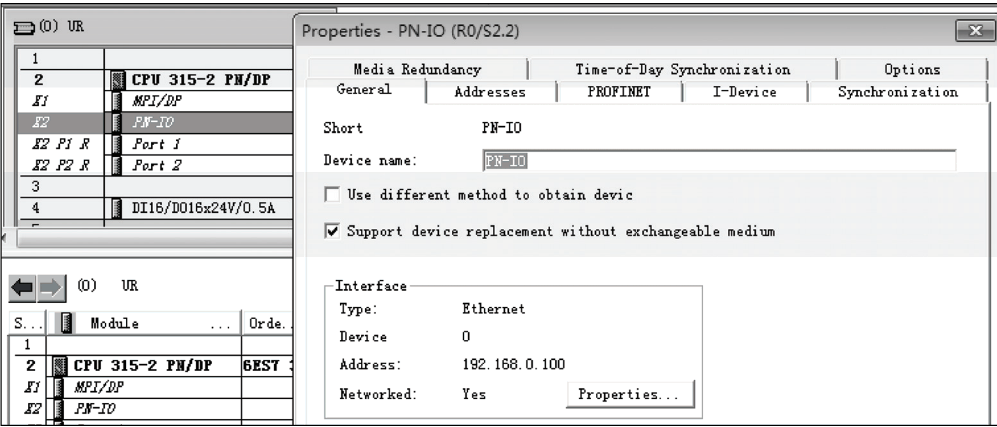


图 8-15　PLC 以太网参数设置

表 8-3　变量分配表

S7 Program(1) (Symbols) -- CNC_Alarm\SIMATIC 300(1)\CPU 315-2 PN/

	Stat	Symbol /	Address		Data type	
1		ALARM	FB	1	FB	1
2		1#缺料按钮	I	0.0	BOOL	
3		1#求援按钮	I	0.1	BOOL	
4		2#缺料按钮	I	0.2	BOOL	
5		2#求援按钮	I	0.3	BOOL	
6		3#缺料按钮	I	0.4	BOOL	
7		3#求援按钮	I	0.5	BOOL	
8		4#缺料按钮	I	0.6	BOOL	
9		4#求援按钮	I	0.7	BOOL	
1		信号复位按钮	I	1.0	BOOL	
1		广播信息	DB	99	DB	99
1		1#ALARM DATA	DB	1	FB	1
1		2#ALARM DATA	DB	2	FB	1
1		3#ALARM DATA	DB	3	FB	1
1		4#ALARM DATA	DB	4	FB	1

其中 DB99 数据块内容为广播报警信息，信息内容见表 8-4。

表 8-4　广播报警内容

LAD/STL/FBD - [DB99 -- "广播信息" -- CNC_Alarm\SIMATIC 300(1)\CPU 315-2 PN/DP\...\DB99]

File Edit Insert PLC Debug View Options Window Help

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	AD01_1	BOOL	FALSE	CNC01缺料
+0.1	AD01_2	BOOL	FALSE	CNC01求援
+0.2	AD02_1	BOOL	FALSE	CNC02缺料
+0.3	AD02_2	BOOL	FALSE	CNC02求援
+0.4	AD03_1	BOOL	FALSE	CNC03缺料
+0.5	AD03_2	BOOL	FALSE	CNC03求援
+0.6	AD04_1	BOOL	FALSE	CNC04缺料
+0.7	AD04_2	BOOL	FALSE	CNC04求援
=2.0		END_STRUCT		

打开 STEP 7 编程软件，建立项目“CNC_Alarm”，项目结构如图 8-16 所示。

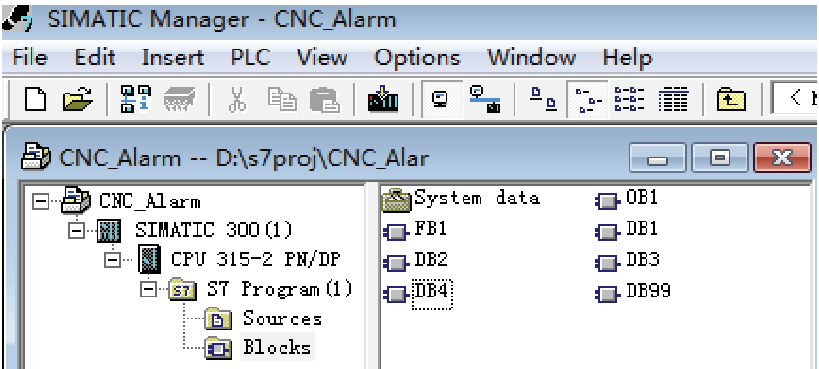


图 8-16　项目结构

8.3.3　程序实现

1. FB1 程序设计

由于 4 台机床报警内容相同，因此建立 FB 模块，编写逻辑关系，每台机床调用 FB1 模块，配上不同的背景数据块即可。

新建 FB1 模块，打开模块，建立接口变量如图 8-17 所示，根据控制要求编写 FB1 逻辑关系如图 8-18 所示。

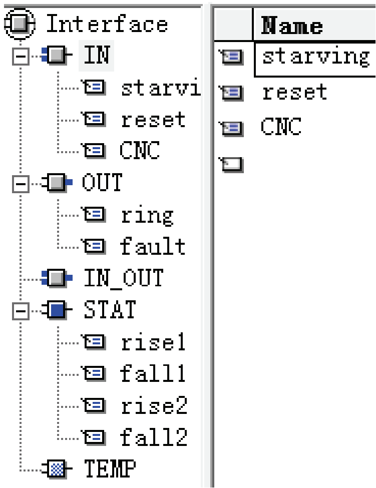


图 8-17　接口变量设置

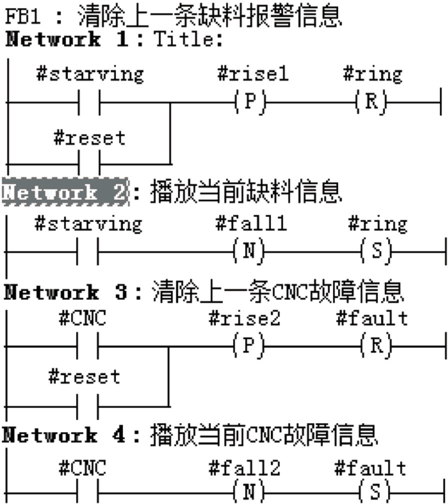


图 8-18　FB1 程序

2. OB1 程序

编写 OB1 程序，每台机床控制调用一次 FB1 模块，将本地输入变量与 FB1 模块输入端连接，将报警变量对应 FB1 模块输出变量，对应背景数据块分别设为 DB1、DB2、DB3 及 DB4，程序调用情况如图 8-19 所示。

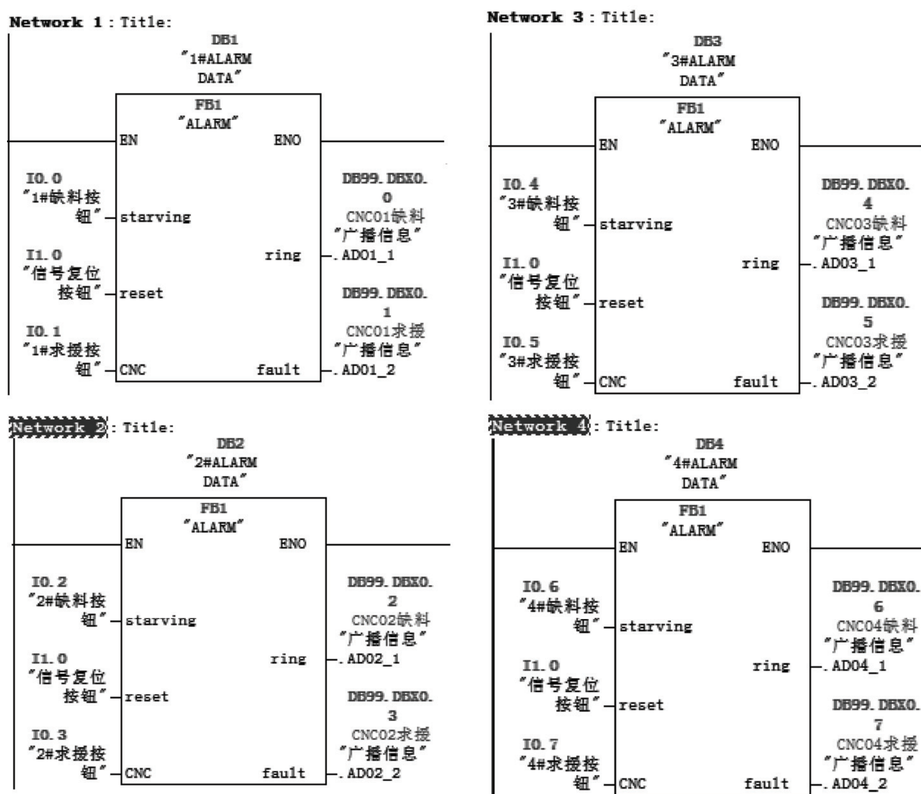


图 8-19　OB1 程序

8.3.4　系统联调

广播系统通过以太网与车间网络连接，并通过 OPC 服务器获得 DB99 数据块信息。OB1 模块在线监控如图 8-20 所示，数据块在线信息如图 8-21 所示。

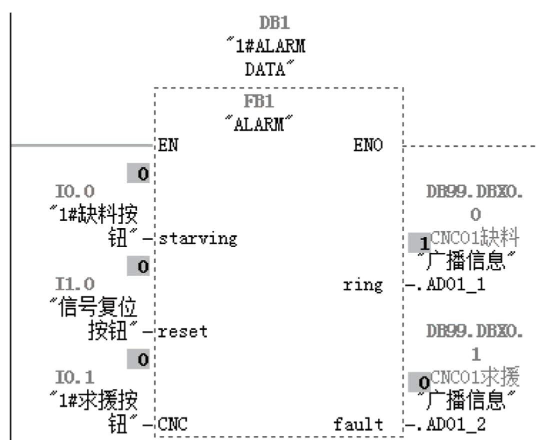


图 8-20　1#CNC 操作台呼叫信息

- "广播信息" -- CNC_Alarm\SIMATIC 300(1)\CPU 315-2 PN/DP\...\DB99 ONLINE]

Debug View Options Window Help

Address	Name	Type	Initial value	Actual value	Comment
0.0	AD01_1	BOOL	FALSE	TRUE	CNC01缺料
0.1	AD01_2	BOOL	FALSE	FALSE	CNC01求援
0.2	AD02_1	BOOL	FALSE	FALSE	CNC02缺料
0.3	AD02_2	BOOL	FALSE	FALSE	CNC02求援
0.4	AD03_1	BOOL	FALSE	FALSE	CNC03缺料
0.5	AD03_2	BOOL	FALSE	FALSE	CNC03求援
0.6	AD04_1	BOOL	FALSE	FALSE	CNC04缺料
0.7	AD04_2	BOOL	FALSE	FALSE	CNC04求援

图 8-21　DB99 数据块在线运行数据

8.4　基于模拟量的液位控制系统设计与实现

8.4.1　控制要求及硬件配置

有一液位控制系统，设计要求如下：

1）按下启动按钮，搅拌系统按照液体 A→液体 B→液体 C→电动机搅拌→放出混合液体的顺序工作。

2）当系统处在液体 A、液体 B 或液体 C 对应阀门工作时，按下停止按钮，当前液体的液位高度完成后才能够停止工作，再按下启动按钮，继续进入下一阶段工作。

3）根据不同的工艺，液体 A、液体 B 及液体 C 注入的液体高度可调，电动机搅拌时间也可调，但必须是本周期工作完成后，才能执行新的工艺。

4）按下急停按钮，系统停止工作。

8.4.2　硬件电路设计

系统主要由上位机、PLC 控制器、被控对象以及液位传感器等部分组成。可以考虑选择 CPU314C-2DP，该模块是紧凑型 CPU，带有集成数字量和模拟量 I/O 以及 PROFIBUS-DP 接口，既能满足本系统对数字量和模拟量的需求，又方便后续的通信功能扩展。

CPU314C-2DP 模块集成有 5 路模拟量输入和 2 路模拟量输出，其模拟量性能见表 8-5。

表 8-5　CPU314C-2DP（订货号：6ES7 314-6CH04-0AB0）模拟量部分技术规范

通道性质	点数	屏蔽电缆最大长度/m	电压输入/输出范围/V	电流输入/输出范围/mA	精度
模拟量输入	5（其中 1 路为热电阻/电阻输入）	100	0~10；−10~10	0~20 −20~+20 4~20	12 位
模拟量输出	2	200			

硬件组态、参数配置及程序的编写由软件 STEP 7 V5.5+SP1 或更高版本、STEP 7 V5.35 + SP2 或更高版本（带 HSP203）完成。

液位传感器选择 DG 系列普及型压力变送器。该类变送器稳定性高，且经过高可靠性的放大电路及温度补偿，将被测介质的表压转换成 DC 4~20mA 标准电信号，可用于液压及气动控制系统、液位测量与控制系统等场合。本系统可根据液体混合容量选择不同量程的传感器，传感器安装位置如图 8-22 所示。

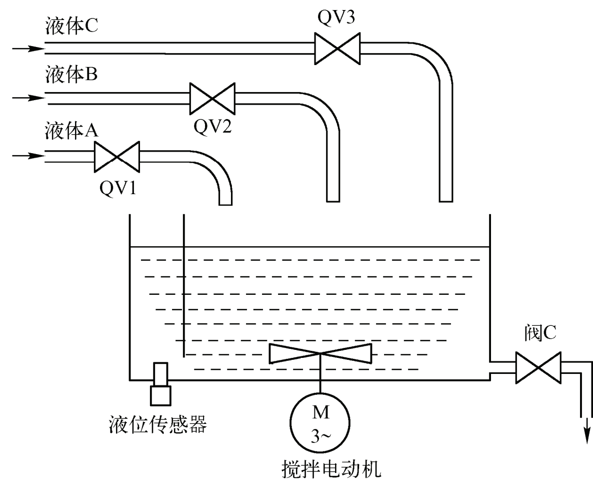


图 8-22　液位混合系统结构

8.4.3　电气原理接线图

根据控制要求，PLC I/O 地址分配及外部接线如图 8-23 所示。

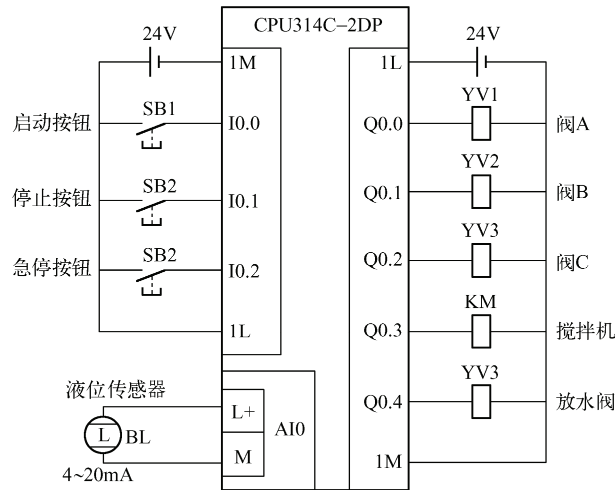


图 8-23　PLC 外部接线图

8.4.4 程序结构

根据控制要求，设计程序结构如图 8-24 所示。由于阀门进水需要重复调用 3 次，故将其设定为 FB 功能块；搅拌和液体放空在每个周期中只调用 1 次，可以使用 FC 功能。

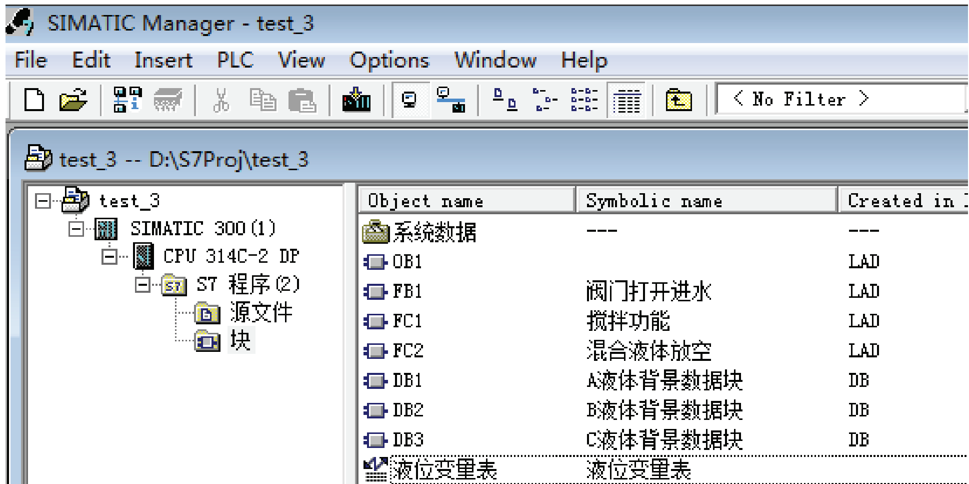


图 8-24　程序结构

在硬件配置界面中，双击“2.3 AI5/AO2”一栏，弹出“Properties”对话框。选择“Addresses”选项卡，将系统默认的模拟量地址修改为：AI→（260~269），AO→（270~273）；在“Inputs”选项卡，将通道 2 性能设为与系统所选液位传感器相对应的参数，如图 8-25 所示。

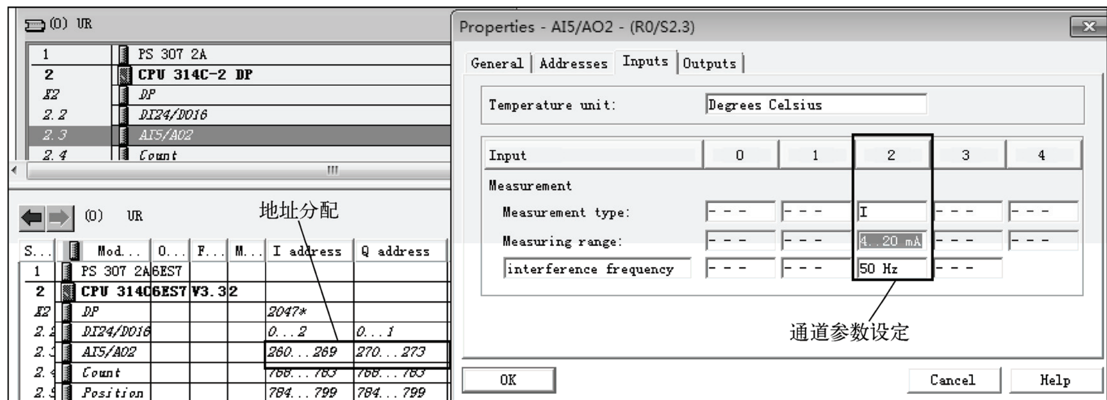


图 8-25　模拟量参数设置

为方便管理及增加程序的可读性，编写系统符号表见表 8-6。

表 8-6　符号表

Symbol /	Address	Data type
A液体背景数据块	DB 1	FB 1
A液体设定高度	MW 4	INT
A液体设定高度测量值	MW 14	INT
B阀门	Q 0.1	BOOL
B液体背景数据块	DB 2	FB 1
B液体设定高度	MW 6	INT
B液体设定高度测量值	MW 20	INT
B液体设定高度数字量	MW 16	INT
C阀门	Q 0.2	BOOL
C液体背景数据块	DB 3	FB 1
C液体设定高度	MW 8	INT
C液体设定高度测量值	MW 22	INT
C液体设定高度数字量	MW 18	INT
VAT_1	VAT 1	
按下停止按钮标志位	M 0.2	BOOL
初始状态	M 0.0	BOOL
单位输入对应的数字量	MW 12	INT
阀门打开进水	FB 1	FB 1
放空标志位	M 0.1	BOOL
放水阀	Q 0.4	BOOL
放水延时	T 2	TIMER
混合液体放空	FC 2	FC 2
急停按钮	I 0.2	BOOL
搅拌功能	FC 1	FC 1
搅拌延时	T 1	TIMER
启动按钮	I 0.0	BOOL
水箱总高度设定值	MW 2	INT
停止按钮	I 0.1	BOOL
液体搅拌	Q 0.3	BOOL
液位死区高度设置	MW 24	INT

8.4.5　功能实现

1．液位高度的计算

液位采集信号进入 PLC 的模拟量输入通道，将该通道参数设置为单极性、分辨率 12 位。假定 H 为水箱高度的最大有效值，压力传感器输出电流 4~20mA，则对应液位高度为（0~100%）H，对应数字量变化范围为（0~27648）；设某一时刻液位高度 h 对应的数字量为 N，则 N 表示的实际液位高度为

$$h=H*N/27648$$

设 A 液体的设定高度是 hA，B 液体的设定高度是 hB，C 液体的设定高度是 hC。则在程序中，B 液体液位高度设定值对应的测量值为（hA + hB），C 液体液位高度设定值对应的测量值为（hA + hB + hC）。

2．FB 的设计

程序中采用比较指令进行液体高度判断，当到达设定液位高度则转入下一段工艺控制，功能实现如图 8-26 所示，背景数据块结构如图 8-27 所示。

其中输入变量 set_height，可以由变量表（或触摸屏）修改给定的液位高度；与输入有关的变量 out0、out1 的作用是：当液位实际高度大于等于设定高度时，复位当前工作状态并切

换到下一工作状态。

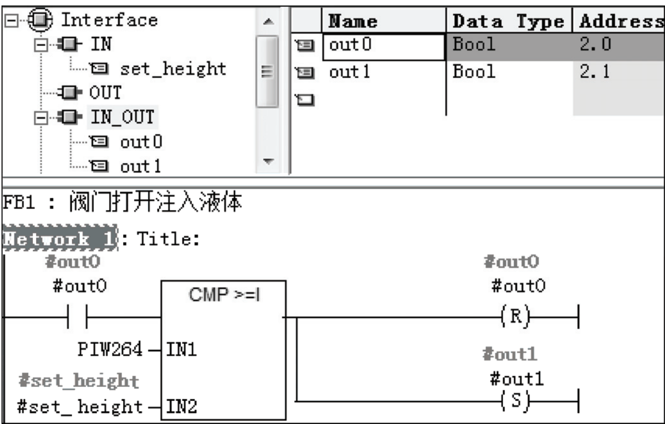


图 8-26　FB1 功能实现

Address	Declaration	Name	Type	Initial value	Actual value
0.0	in	set_height	INT	0	0
2.0	in_out	out0	BOOL	FALSE	FALSE
2.1	in_out	out1	BOOL	FALSE	FALSE

图 8-27　FB1 的背景数据块

3．搅拌功能 FC1

液体进行混合搅拌时，需要对搅拌动作和搅拌时间进行控制，FC 模块为无参数设计，功能实现如图 8-28 所示。

4．混合液体放空功能 FC2

放水阀打开，混合液体流出，当液位检测值等于设定死区液位高度时，比较指令满足条件，M0.1=1，放水阀继续工作 5s，将残留液体放出，功能实现如图 8-29 所示。

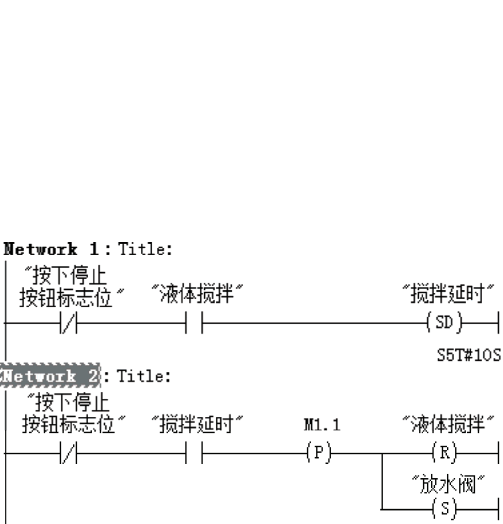


图 8-28　搅拌功能程序

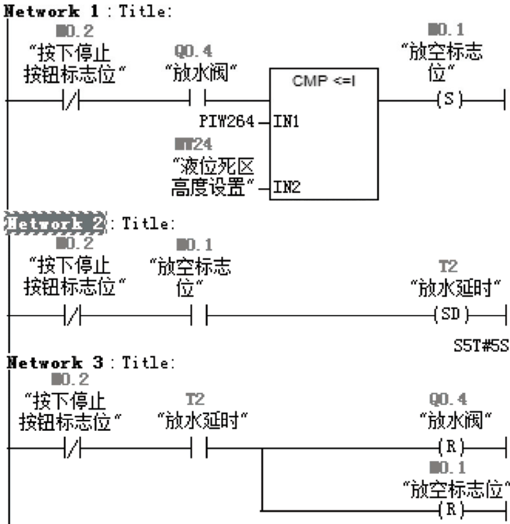


图 8-29　混合液体放空功能实现

5．OB1 程序设计

OB1 程序完成系统初始化（例如液位高度计算、液位高度设置）功能，切换系统工作模式（例如启动、停止、急停），调用 FB、FC 功能，使得系统按照设计要求有序地工作，具体实现程序如图 8-30 所示。

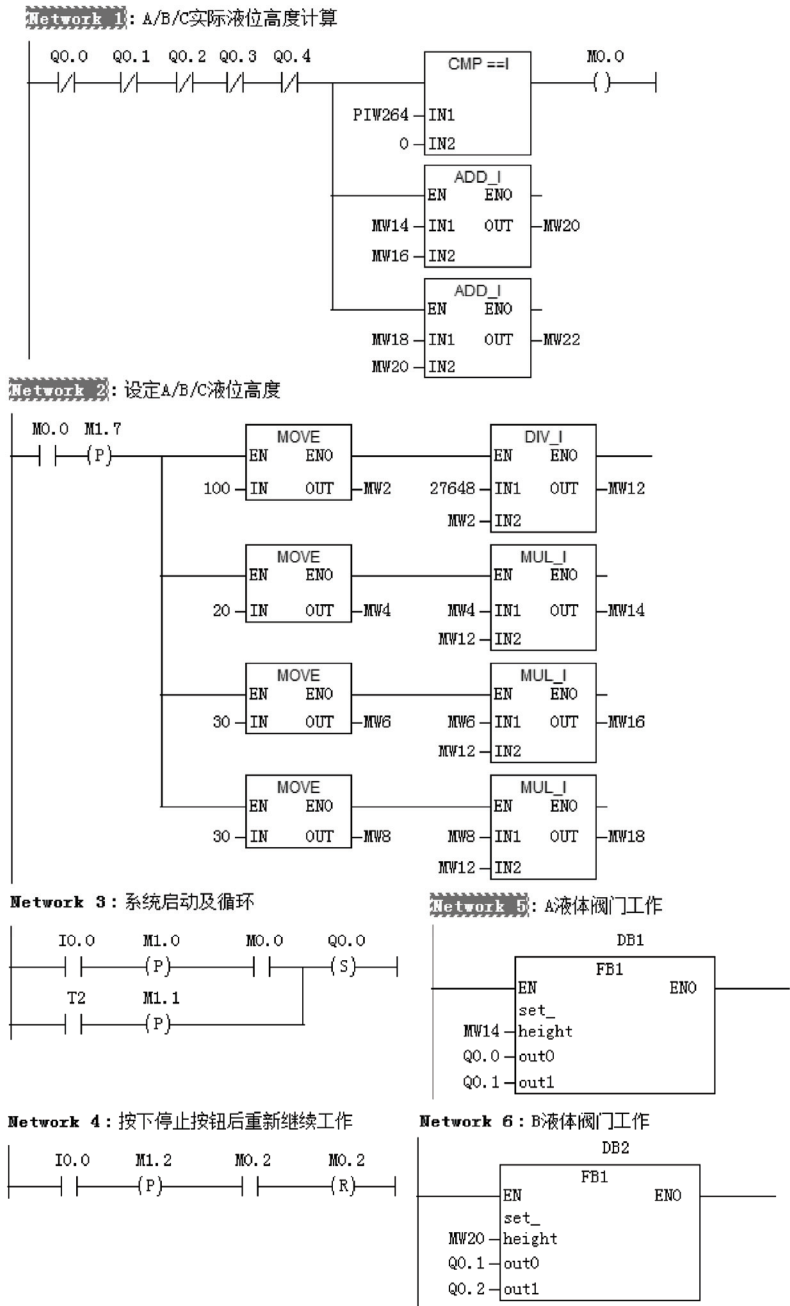


图 8-30　OB1 程序

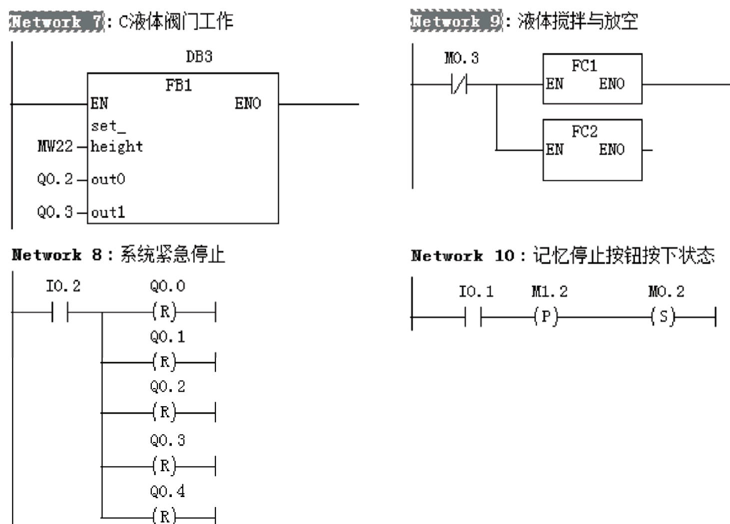


图 8-30　OB1 程序（续）

8.5　基于 PID 的位置控制系统设计与实现

8.5.1　项目背景

在高校实验室中，目前使用的 PID 控制实训载体多采用基于温度、液位调节的控制系统，该类载体在实际应用中，一是对实训环境要求较高，如需配置给排水管路、增加电气负荷、操作存在安全隐患等；二是该类控制系统多为大惯性环节，参数设定困难，调节周期长；三是实训准备耗时长，故障率高，维修困难。

因此本项目设计了一套基于 PID 功能的实训系统，控制对象为玻璃管内的乒乓球。由于该控制对象为小惯性环节，调节时间短，曲线理想，非常适合学生在有限的实验实训时间内训练。

8.5.2　硬件选型

实训载体由支架、轴流风机、透明亚克力玻璃管、乒乓球等构成。控制系统由 PLC、HMI、PWM 板、电位器、激光测距仪、轴流风机、透明亚克力玻璃管和乒乓球组成，其电气原理如图 8-31 所示，通过测距仪反馈兵乓球位置，调节风机的速度，使得小球的位置稳定在设定高度。

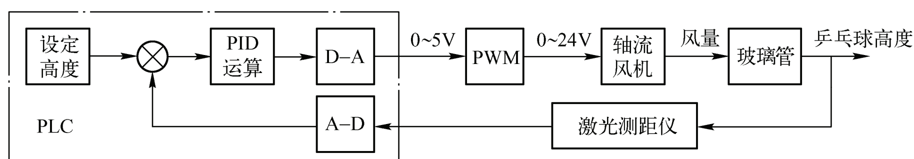


图 8-31　电气原理图

考虑选择 CPU314C-2DP，该模块是紧凑型 CPU，带有集成数字量和模拟量 I/O，能满足本系统对数字量和模拟量的需求。

由于该系统 PLC、HMI 需要通过以太网与整个实训网络通信，可以选择 CP343-1 以太网通信模块，以便 PLC 与 HMI 及 PC 等通过以太网通信。

HMI 选择 MP277 10' Touch 型号，采用 PROFINET 接口与 PLC 通信。

激光测距仪型号 XUE 5AA2NM12，输出信号 4~20mA。

PLC 控制器输出信号 0~5V，用来控制 PWM 的脉冲宽度，从而控制风机风量。

8.5.3 电气原理接线图

根据控制要求及以上元器件选型，设计系统电气原理图如图 8-32 所示。

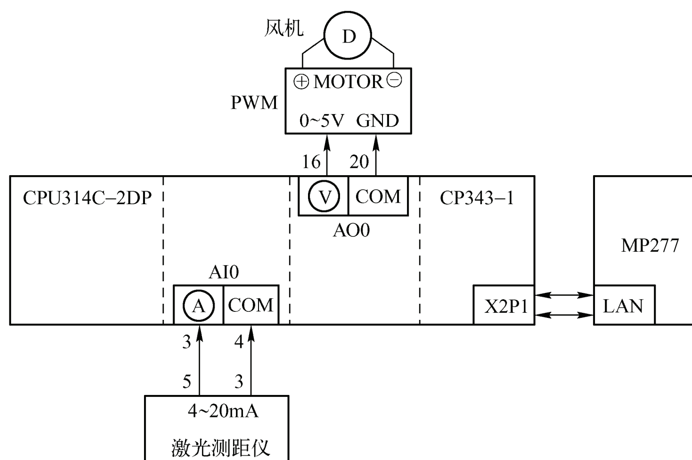


图 8-32 电气原理图

8.5.4 项目结构及硬件组态要点

1）打开编程软件 STEP 7 V5.5，建立新项目“ping-pong”，并将 HMI 项目集成在该项目中，项目结构如图 8-33 所示。

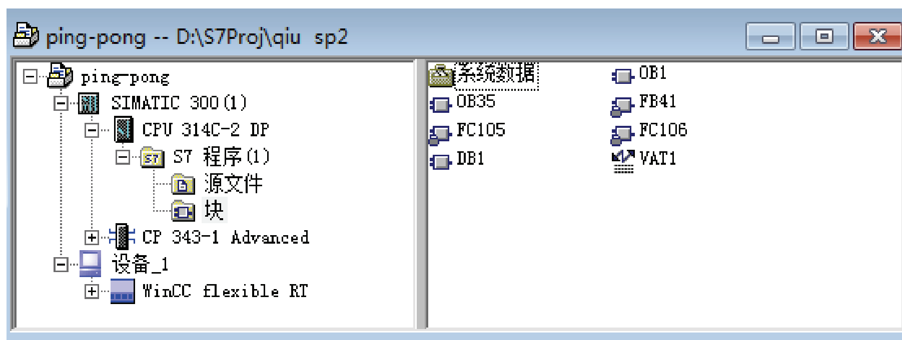


图 8-33 项目结构

2）在硬件组态中，激光测距仪输入通道参数设置如图 8-34 所示。

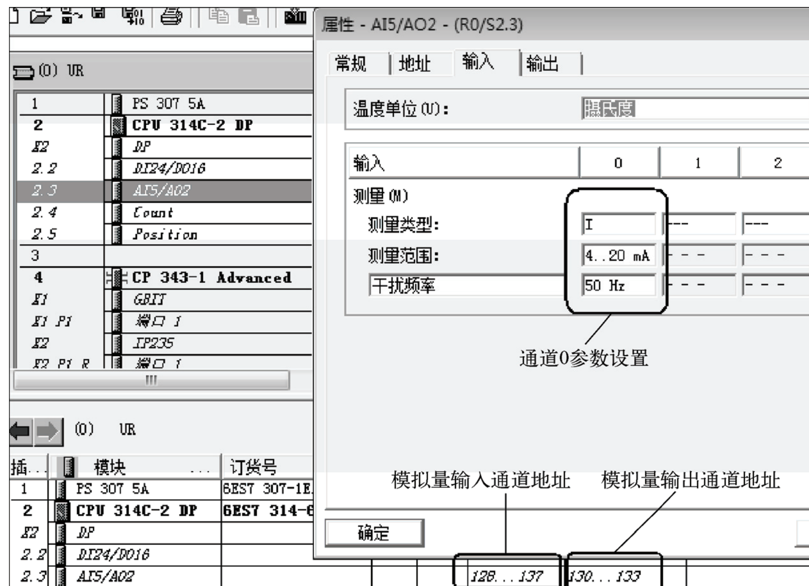


图 8-34　模拟量输入通道参数设置

输出通道参数设置如图 8-35 所示。

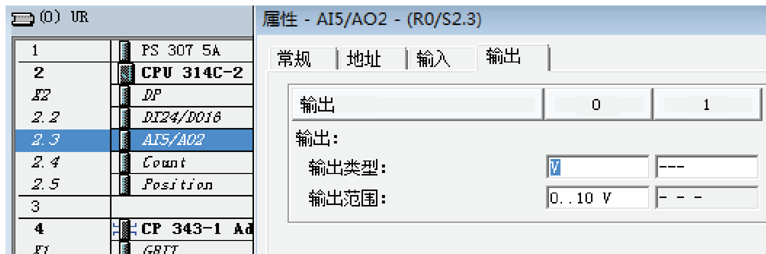


图 8-35　模拟量输出通道参数设置

双击机架中的 CPU，选择“循环中断”选项卡，如图 8-36 所示，修改 OB35 执行周期（1~60000ms）为 20ms，硬件组态完毕后，编译并保存硬件组态文件。

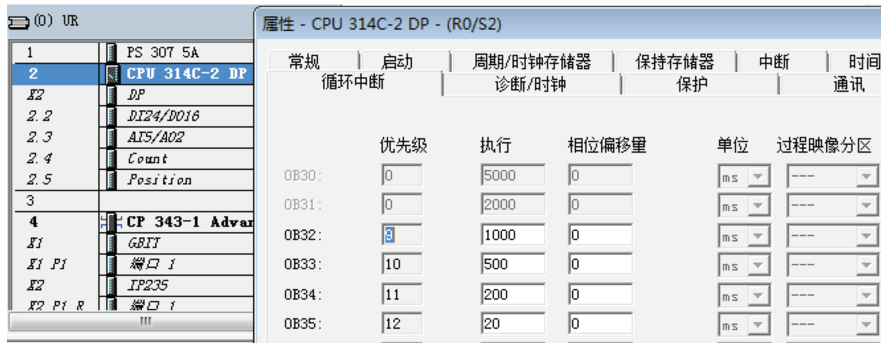


图 8-36　OB35 组织块参数设置

3）HMI 项目的集成。打开 WinCC flexible 软件，创建一个空项目→单击确定→选择设备类型→单击确定，创建成功。详细步骤如图 8-37 和图 8-38 所示。

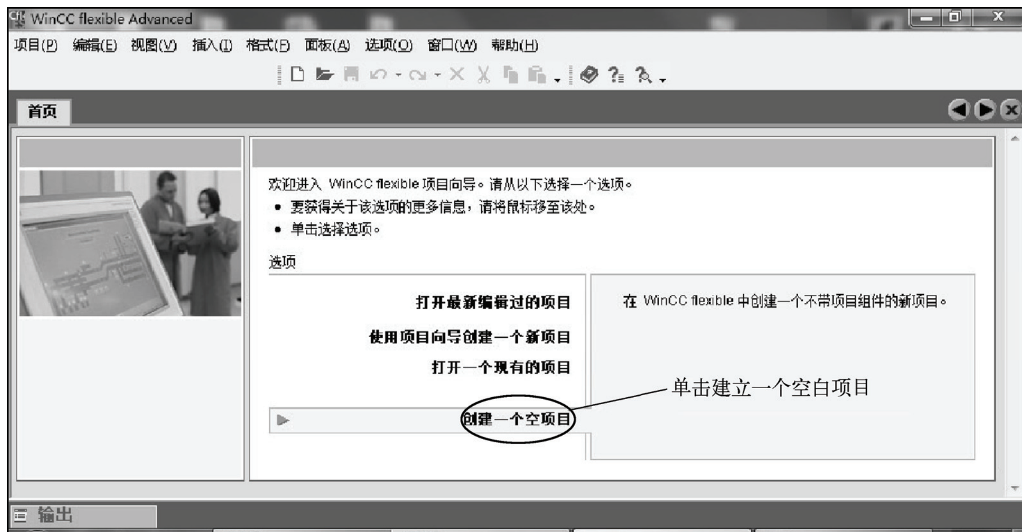


图 8-37　创建一个新项目



图 8-38　选择 HMI 型号

单击图 8-38“确认”按钮，空白新项目建成。在“项目”下拉菜单中单击“在 STEP 7 中集成…”选项，弹出图 8-39 界面，选择在 STEP 7 软件中已建立的 PLC 工程项目，单击“确认”按钮，将该 HMI 工程集成到 STEP 7 管理界面中，并能从管理界面直接进入 WinCC flexible 软件中。



图 8-39　将 HMI 工程集成在 STEP 7 项目中

8.5.5　PID 功能

在 PLC 中，可以自己编写程序，实现 PID 运算功能，也可以调用 PLC 中提供的 PID 模块，按照模块说明设置相关参数，本例采用 PID 功能块实现系统闭环控制功能，该类功能块的性能和处理速度只取决于所使用的 CPU 性能。

PID 控制包的功能块包含了用于连续控制的 CONT_C（FB41）功能块、用于步进控制的 CONT_S 功能块以及用于脉宽调制的 PULSEGEN 的功能块。控制器块实现了一个纯软件控制器，它所包含的功能块可以提供控制器的所有功能，循环计算所需要的数据存储在为 FB 分配的数据块中，这样可以按照需要频繁地调用 FB。

小球位置和风机转动属于连续输入和输出变量的工艺过程，因此采用 FB41 功能块。

PID 功能块调用路径如图 8-40 所示，FB41 功能块设置参数如图 8-41 所示。

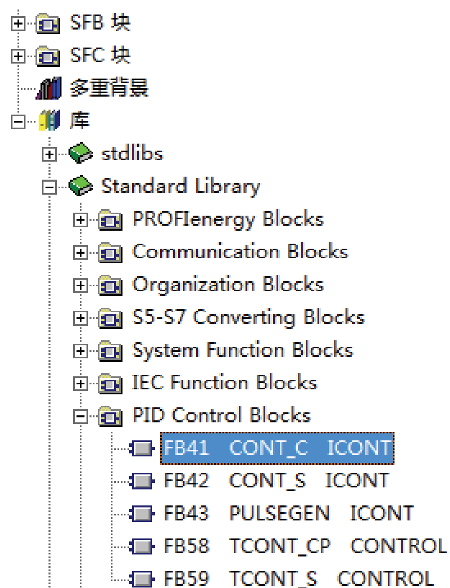


图 8-40　FB41 的位置

图 8-41 为 FB41 功能块需要设置的参数，在这里仅对本例需要用到的参数做一个简单说明。

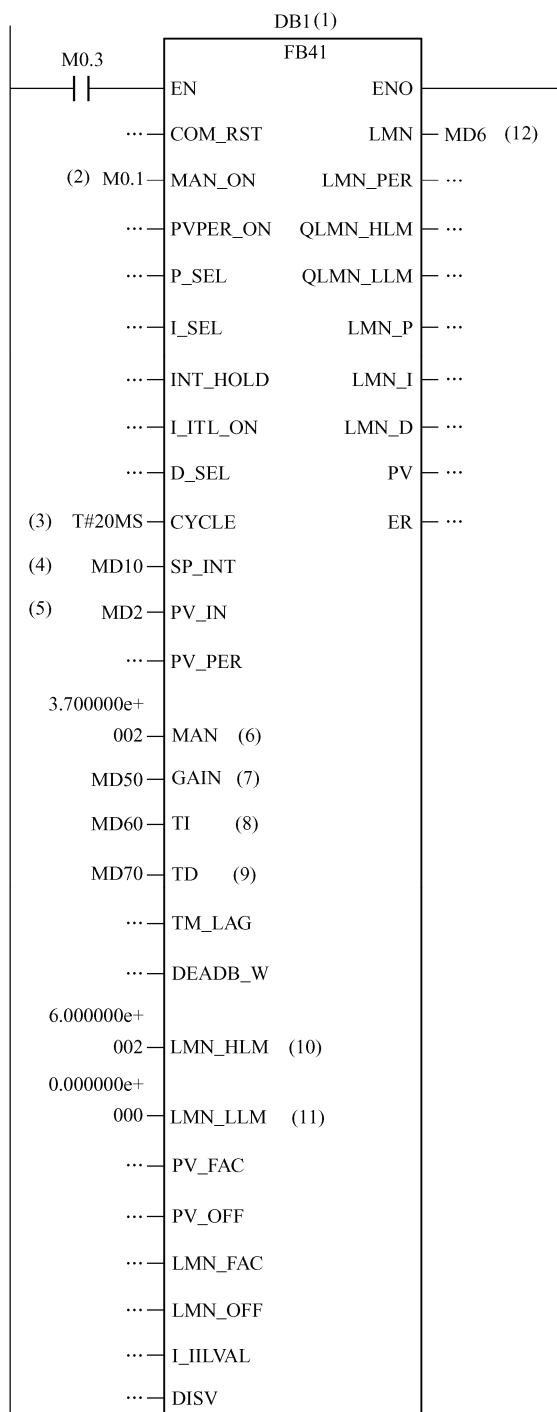


图 8-41　FB41 参数设置

图 8-41 为 FB41 功能块需要设置的参数，在这里仅对本例需要用到的参数做一个简单说明。

1）DB1：功能块 FB41 的背景数据块。

2）MAN_ON：手动值打开，当该位为“1”时，PID 可直接将 MAN 的值输出到 LMN。

3）CYCLE：采样时间，块调用之间的时间间隔必须恒定，采样时间输入确定了块调用之间的时间间隔。

4）SP_INT：内部设定值，用于输入小球高度的设定值。

5）PV_INT：过程变量输入值，用于实时显示小球当前高度值。

6）MAN：手动值，当 MAN_ON=1 时有效。

7）GAIN：比例增益。

8）TI：积分时间。

9）TD：微分时间。

10）LMN_HLM：调节值上限，调节值总是受到上限和下限的限制，该值指定调节值的上限。

11）LMN_LLM：调节值下限，调节值总是受到上限和下限的限制，该值指定调节值的下限。

12）LMN：PID 计算后的输出值。

8.5.6 触摸屏界面设计

为了便于操作和管理，设计 HMI 界面时将界面分为 3 页，分别是进入界面、操作界面和监视界面。

1．进入界面

进入界面是 HMI 首页，用于显示项目名称、进入监控界面及退出系统，运行效果如图 8-42 所示。



图 8-42　进入界面

设计要点如下：

（1）背景画面的导入

如图 8-43~图 8-45 所示，在右侧工具栏中，单击图形→选择“我的图像文件夹”→右键单击，选择“新文件夹链接”，添加需要的图片。



图 8-43　建立图片文件夹



图 8-44　选择图片

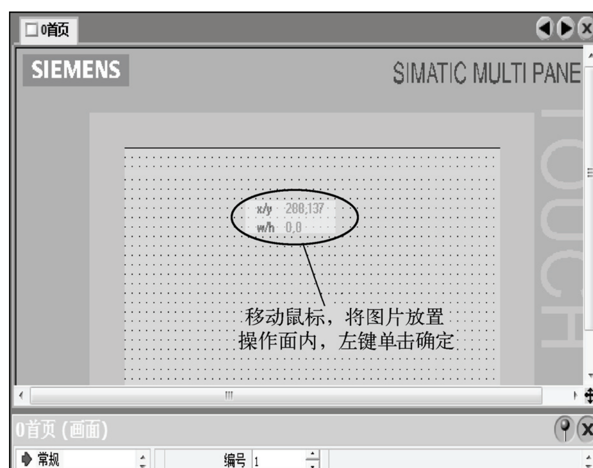


图 8-45　图片位置

双击图片，弹出设置框如图 8-46 所示，选择“属性”选项，修改“布局”参数。

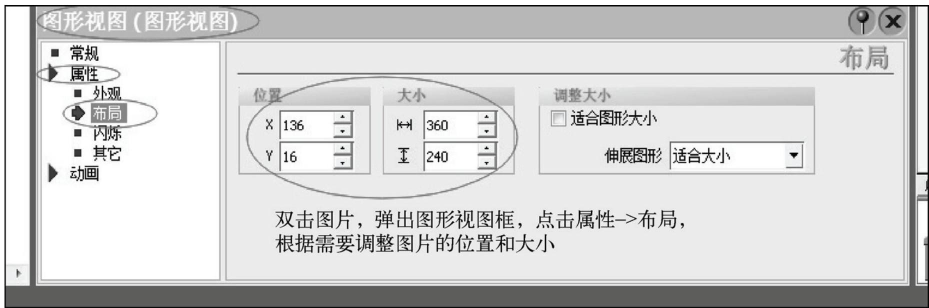


图 8-46 “属性”参数设置

（2）加入时钟显示功能

如图 8-47 所示，时钟进入路径：工具栏→增强对象→时钟，双击时钟，设置时钟参数。

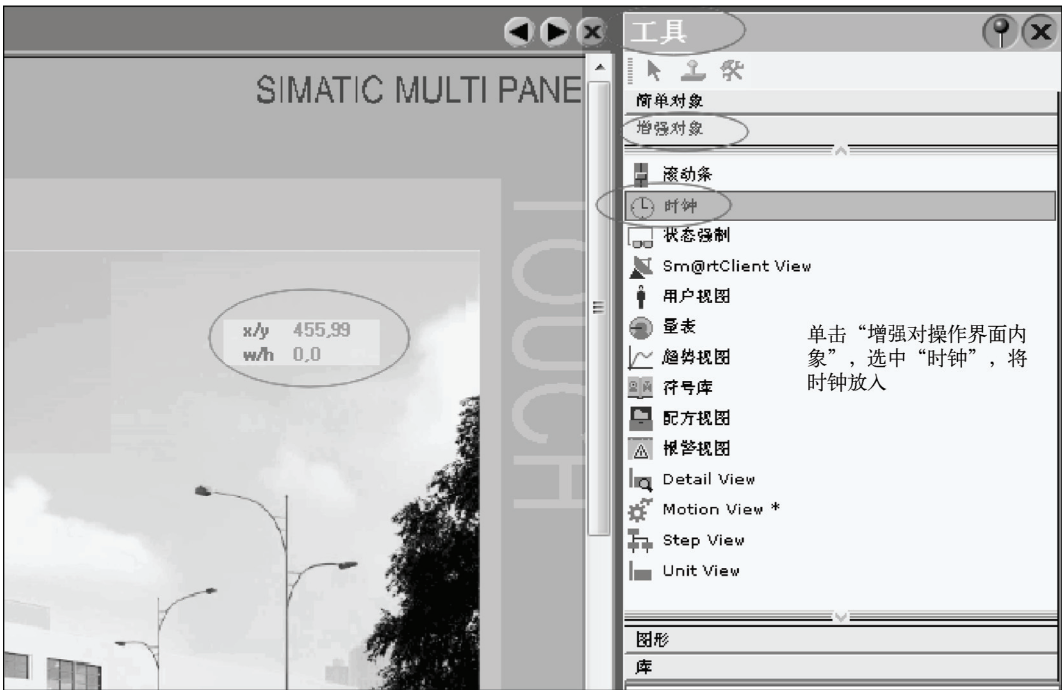


图 8-47 时钟画面制作

（3）添加文本

如图 8-48 所示，进入路径：工具栏→简单对象→文本域，添加文本并设置文本属性。

（4）启动按钮设置

如图 8-49 所示，进入路径：工具栏→简单对象→按钮，设置常规、属性、动画等参数，如图 8-50 所示的“动画”的外观栏参数设置。

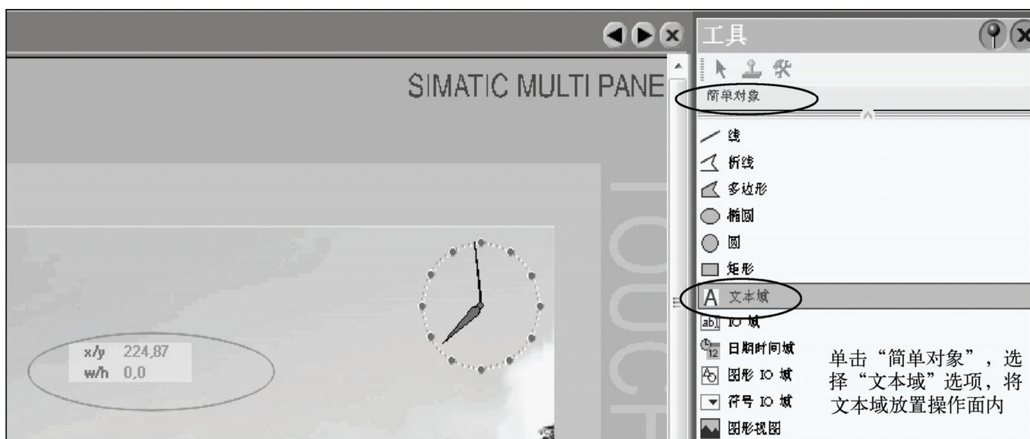


图 8-48　文本画面制作

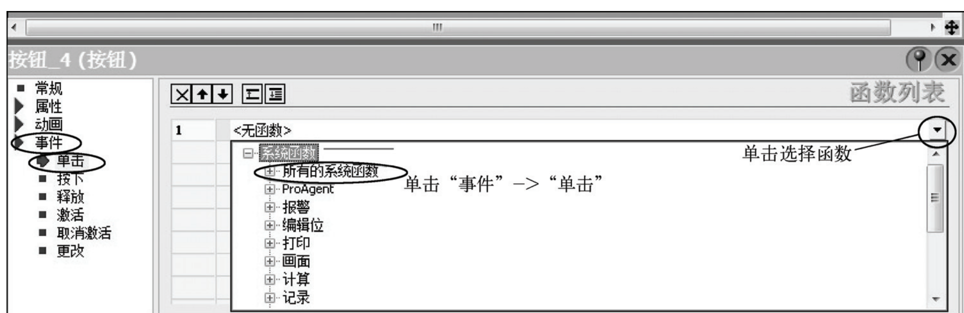


图 8-49　按钮画面制作

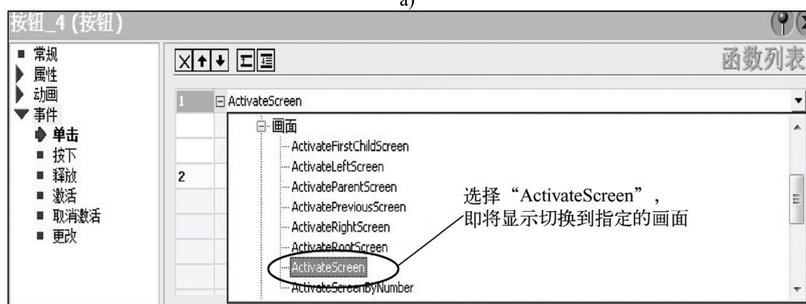


图 8-50　“外观”参数设置

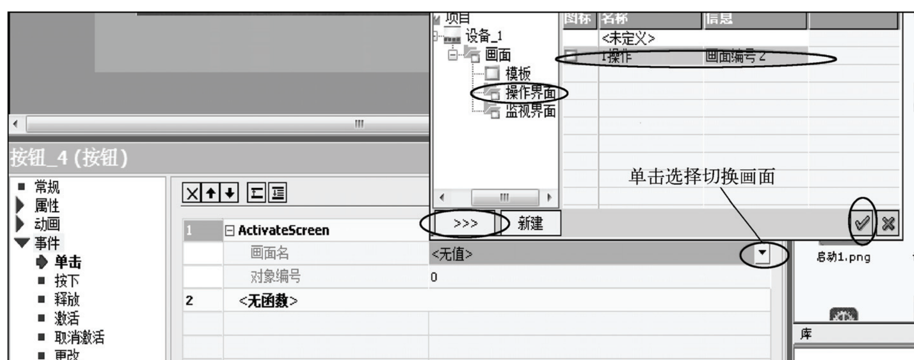
设置按钮按下的事件属性，如图 8-51 所示。



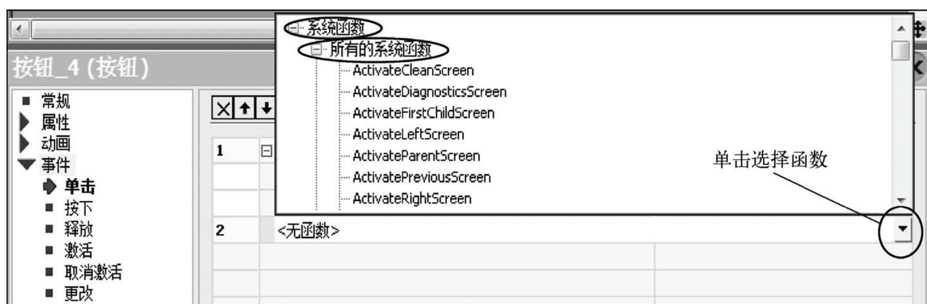
a)



b)

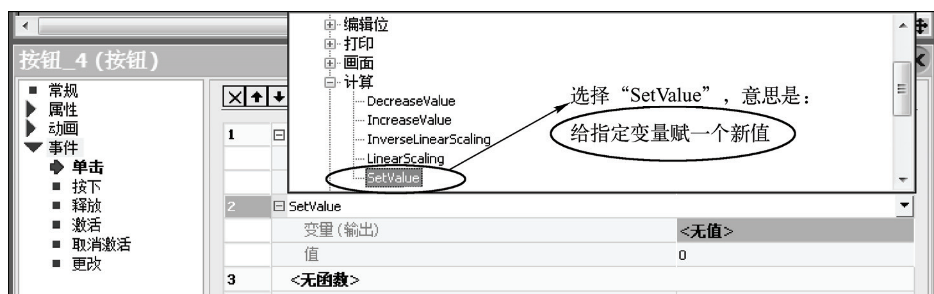


c)

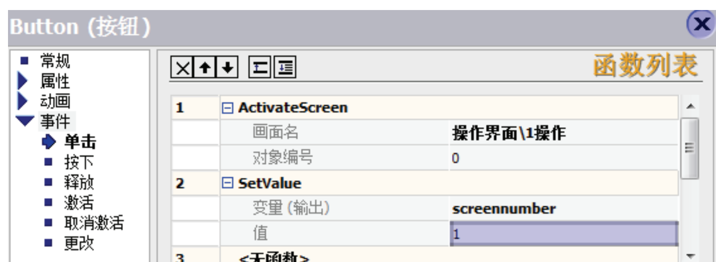


d)

图 8-51 “事件”参数设置



e)



f)

图 8-51 “事件”参数设置（续）

（5）系统按钮设置

系统按钮用于进入触摸屏操作系统，设置路径同（4）“启动按钮设置”。“事件”设置如图 8-52 所示。单击“事件”，选择“系统”→“OpenControlPanel”选项，即打开 WinCE 控制面板窗口。

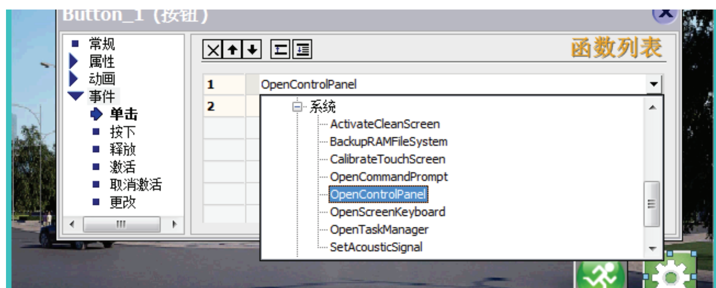


图 8-52 系统按钮设置

2．操作界面

操作界面用于操作、修改系统参数，效果如图 8-53 所示。

设计要点如下：

（1）移动小球

系统设计小球的移动范围为 0~600mm，移动小球用于显示小球当前的设定值。该动画主要是将小球图片与 PLC 程序中小球位置设定变量连接，以便直观查看小球移动位置。动画设置界面如图 8-54 所示。

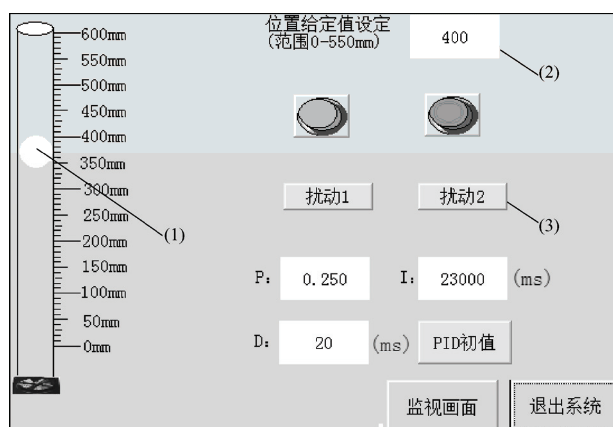


图 8-53　操作界面



图 8-54　小球位置显示参数设置

（2）小球给定值设置

小球给定值设定可以在 PLC 中设定，也可以在 HMI 中随时修改参数，在 HMI 上该变量与 PLC 中的位置给定变量连接。给定值进入路径：工具栏→简单对象→IO 域，如图 8-55 所示；参数设置如图 8-56 所示。

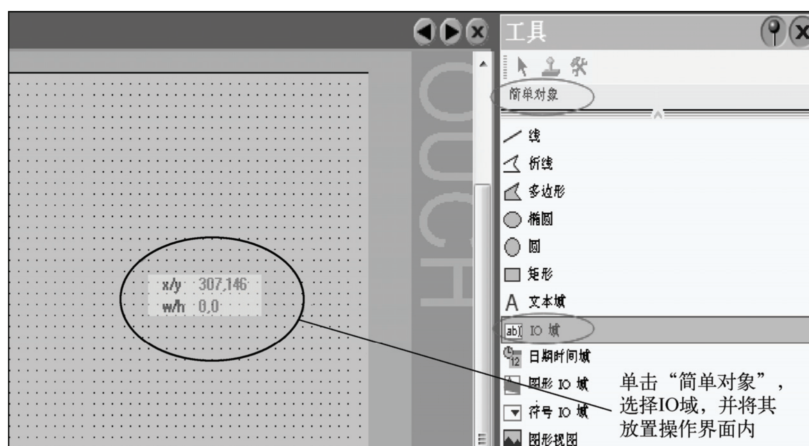


图 8-55　给定值画面制作

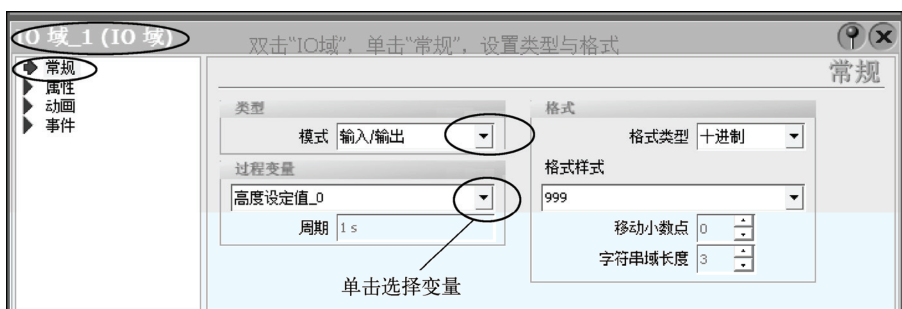


图 8-56　给定值参数设置

（3）扰动设置

扰动设置用于增加或减小风机外力，观察系统在外部扰动作用下系统变化情况，该变量来自于 PLC 程序。画面采用按钮制作，参数设置如图 8-57 所示。



图 8-57　扰动按钮参数设置

3．监视界面

监视界面用于观察小球位置实时变化曲线，界面运行效果如图 8-58 所示。

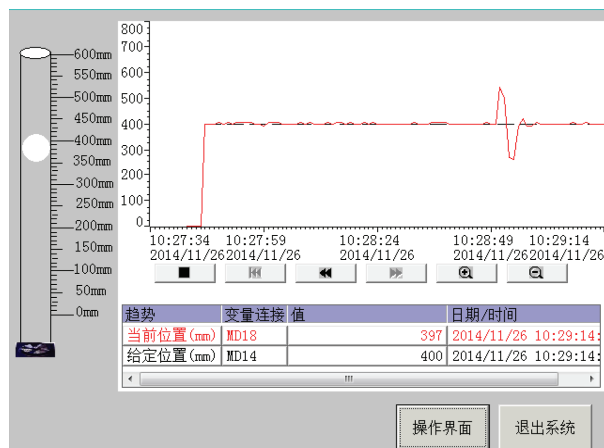


图 8-58　监视界面

设计要点如下：

（1）动态曲线的制作

动态曲线用于显示小球位置随着时间的变化情况。添加曲线路径：工具栏→增强对象→

趋势视图，如图 8-59 所示。

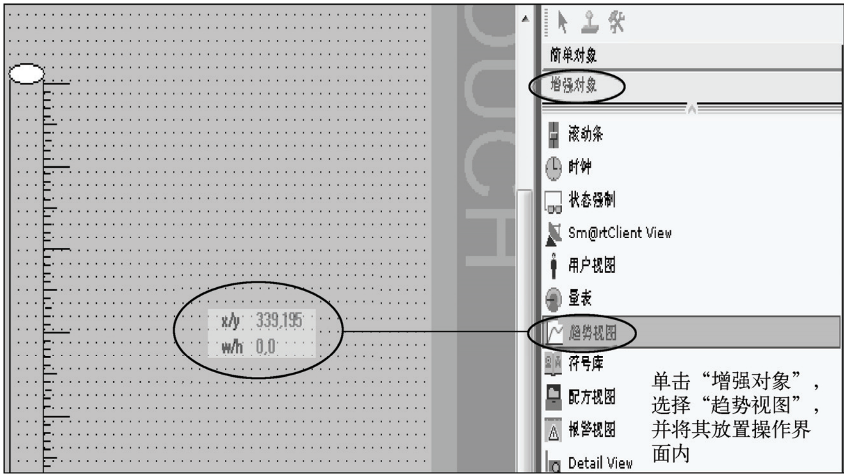


图 8-59 趋势视图制作路径

对趋势图进行“常规”参数设置，如图 8-60 所示。设置趋势图“属性”如图 8-61 所示。

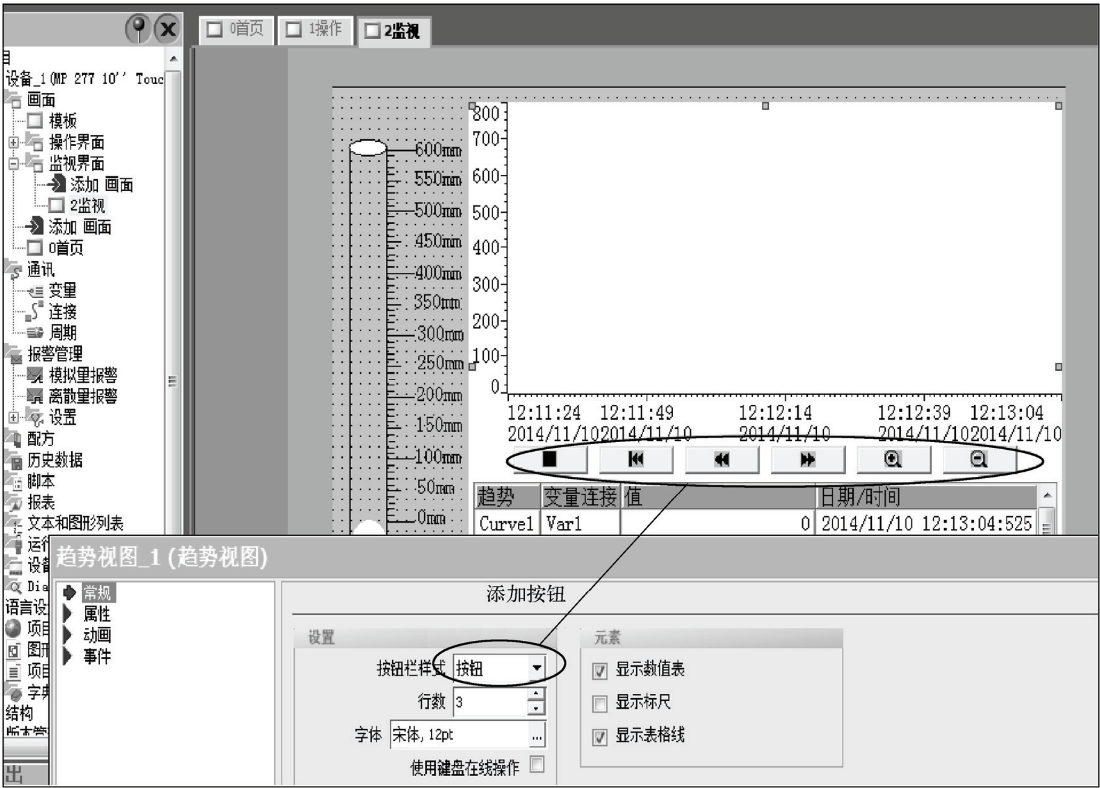
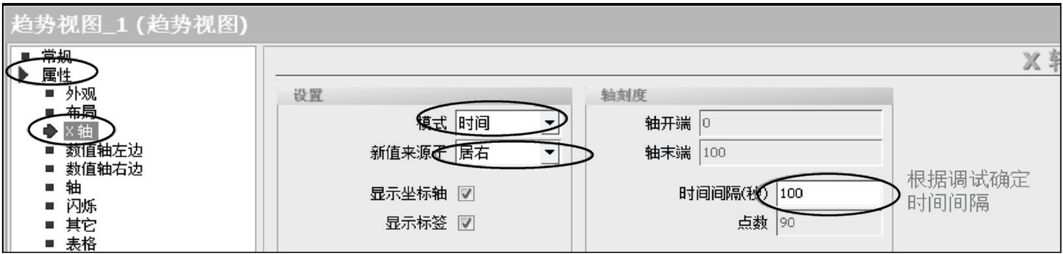


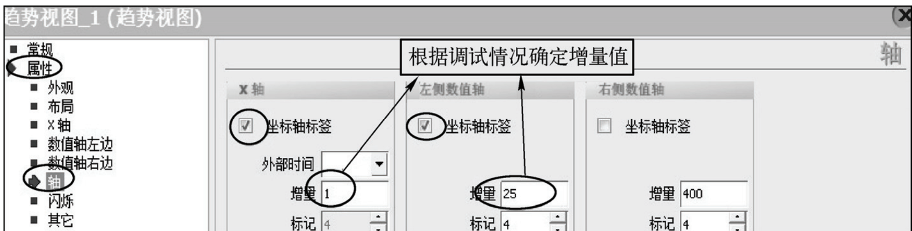
图 8-60 “常规”参数设置



a)



b)



c)



d)

图 8-61　趋势图属性设置

a）“X 轴”参数设置　b）“数值轴左边”参数设置　c）“轴”参数设置　d）“趋势”参数设置

（2）退出系统制作

退出系统按钮用于将画面切换至 HMI 进入画面。按钮性能设置如图 8-62 所示。

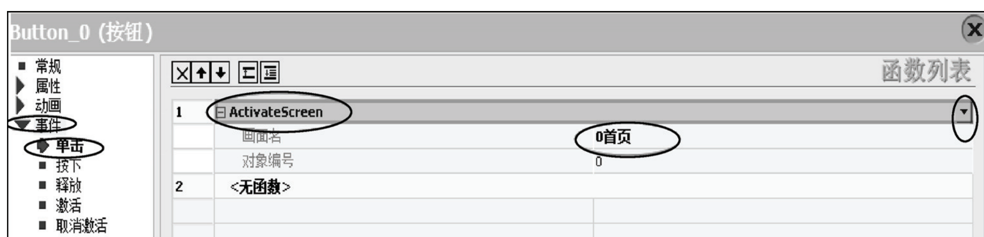


图 8-62 “事件”参数设置

8.5.7 系统功能的实现

在设计 PLC 程序时，设计要点如下：

1）为了保证精度，一般将 FB41 放在 OB35 中。OB35 是个以固定时间间隔循环执行的组织块，间隔时间 20ms 已在硬件配置中设置，使得系统每隔 20ms 执行一次 PID 功能块。

2）由于 HMI 要修改设定位置值和实时显示当前位置值，需要进行模拟量输入、模拟量输出量程的转换，可采用 FC105/FC106 功能。

FC105 将一个整形数（IN）转换成一个在上下限之间变化的实际工程值，并将结果写到 OUT 中。将模拟量输入的数字量转换成工程量如图 8-63 所示。

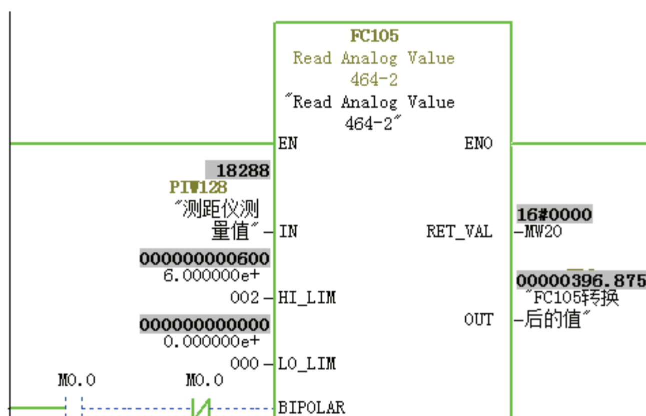


图 8-63 FC105 的应用

在图 8-63 中，HI_LIM 为工程量标定上限、LO_LIM 为工程量标定下限。BIPOLAR 为极性输入端，0 表示输入为单极性，即输入的整形范围为（0~27648），对应工程值范围（HI_LIM~LO_LIM）；1 表示输入为双极性，即输入的整形范围为（-27648~+27648），对应工程值范围为（HI_LIM~LO_LIM）；OUT 为量程转换结果，为实数。

PIW128 是测距仪测量的乒乓球高度，为要转换为工程量的输入值；系统标定乒乓球运动范围为 0~600mm；因此输入值通过 FC105 转化成 0~600mm 范围内变化的工程值。

FC106 将一个实数（IN）转换成上下限之间变化的实际工程值，并将结果写到 OUT 中。将模拟量输出的数字量转换成工程量如图 8-64 所示。

PID 模块按照工程量单位计算后的输出值放在 MD6 中，模拟量输出通道设定为电压、单极性 0~10V 范围；由于 PWM 控制极电压范围为 0~5V 电压，对应高度 0~600mm，为了保证

模拟量输出 0~5V 的电压，因此在标定时 HI_LIM 为 1200，LO_LIM 为 0。

由于工程换算范围为（HI_LIM~LO_LIM），则上限 1200 对应数字量 27648，对于模拟量输出通道而言得到的电压是 10V；下限 0 对应数字量 0，对于模拟量输出通道而言得到的电压是 0V；对于 600 而言得到的输出电压是 5V。

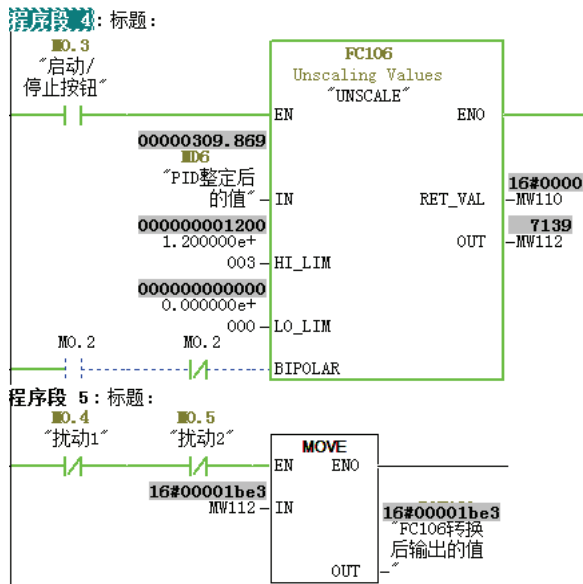


图 8-64　FC106 的应用

第 9 章　S7-300 PLC 的通信

9.1　MPI 通信协议及其应用

9.1.1　MPI 通信协议

MPI 是多点接口（Multi Point Interface）的简称，该接口集成在每个 SIMATIC 的可编程序控制器上（SIMATIC S7/M7/C7）。MPI 物理接口符合 PROFIBUS RS-485（EN 50170）接口标准，网络通信速率为 19.2kbit/s~12Mbit/s，S7-200 PLC 只能选择 19.2kbit/s 的通信速率，S7-300 PLC 通常默认速率是 187.5kbit/s，只有能够设置为 PROFIBUS 接口的 MPI 网络才能支持 12Mbit/s 的通信速率。

MPI 接口的基本功能是作为 S7 的编程接口，还可以实现对通信速率要求不高的、小数据量 PLC 站之间的通信，以及连接 OP/TP，通信数据包不大于 122KB；MPI 通信利用 PLC 站和上位机（PG/PC）插卡 CP5411/CP5412/CP5511/CP5512/CP5611/CP5613 的 MPI 进行数据交换；连接电缆为 PROFIBUS 电缆，接头为 PROFIBUS 接头，并带有终端电阻；连接时需要在 MPI 网络的第一个节点及最后一个节点将终端电阻接入通信总线中，在网络中添加节点时，要切断 MPI 网络电源，网络地址为 0~15。

如图 9-1a 所示，接入到 MPI 网络的设备称为网络节点，不分段的 MPI 网络（无中继器）最多可以有 32 个网络节点，最大通信距离为 50m，即第一个节点到最后一个节点最长距离仅为 50m；如图 9-1b 所示，对于一个要求较大区域的信号传输或分散控制的系统，采用两个中继器可以将两个节点的距离增大到 1000m，通过 OLM 光纤距离可扩展到 100km 以上，但两个节点之间不应再有其他节点。

仅用 MPI 接口构成的网络，称为 MPI 分支网（简称 MPI 网络），两个或多个 MPI 分支网，用网间连接器或路由器连接起来，就能构成较复杂的网络结构，以实现更大范围的设备互连。

每个 MPI 分支网上都有一个分支网络号，以区别不同的 MPI 分支网。分支网上的每个节点都有一个网络地址，称为 MPI 地址，节点的 MPI 地址号不能大于给出的最高 MPI 地址，这样才能使每个节点都能正常通信。

通过 MPI 可实现 S7 PLC 之间三种通信方式：全局数据包通信、无组态连接通信和组态连接通信。S7-200 PLC 只支持无组态连接的 MPI 通信。

9.1.2　全局数据包通信

全局数据（GD）通信方式利用全局数据可以建立分布式 PLC 间的通信联系，以这种通信方式实现 PLC 之间的数据交换时，只需要关心数据的发送区和接收区，在配置 PLC 硬件的过程中，组态所要通信 PLC 站之间的发送区和接收区，不需要任何程序处理，适用于 S7-300/400 PLC 之

间的相互通信，最多可以在一个项目中的 15 个 CPU 之间建立全局数据通信。

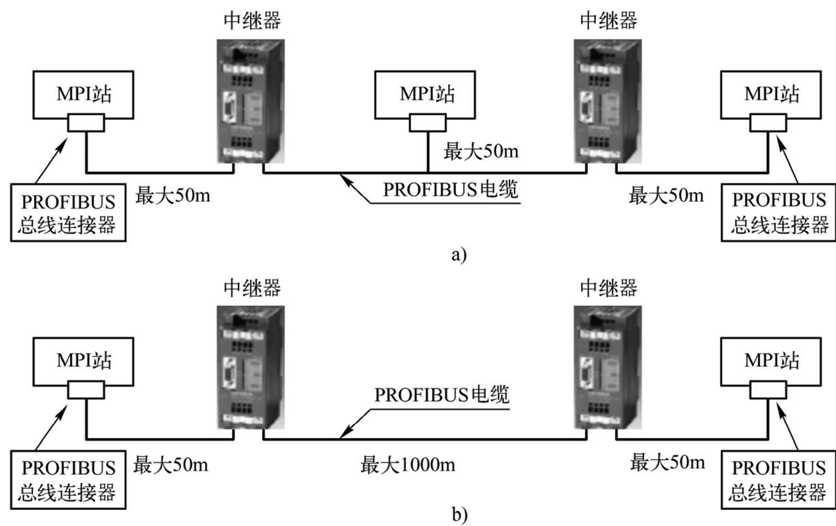


图 9-1 MPI 网络

a) MPI 网络-1 b) MPI 网络-2

1．通信原理

在 MPI 分支网上实现全局数据共享的两个或多个 CPU 中，至少有一个是数据的发送方，有一个或多个是数据的接收方；发送或接收的数据称为全局数据；具有相同 Sender/Recevicer 的全局数据，可以集合成一个全局数据包一起发送；每个数据包用数据包号码（GD Packet Number）来标识，其中的变量用变量号码（Variable Number）来标识；参与全局数据包交换的 CPU 构成了全局数据环（GD Circle），每个全局数据环用数据环号码（GD Circle Number）来标识。

在 PLC 操作系统作用下，发送方的 CPU 在它的一个扫描循环结束时发送全局数据，接收方在它的扫描循环开始时接收数据，即发送全局数据包中的信号状态会自动影响接收数据包；接收方对接收数据包的访问，相当于对发送方数据包的访问。

2．数据结构

全局数据可以由位、字节、字、双字或相关数组组成，它们被称为全局数据的元素。一个全局数据包由一个或几个 GD 元素组成，最多不能超过 24B，见表 9-1。

表 9-1 数据结构

数 据 类 型	类型所占存储字节数	在 GD 中类型设置的最大数量
相关数组	字节数+两个头部说明字节	一个相关的 22 个字节数组
单独的双字	6B	4 个单独的双字
单独的字	4B	6 个单独的双字
单独的字节	3B	8 个单独的双字
单独的位	3B	8 个单独的双字

3．采用 MPI 网络实现 S7-300 PLC 之间的通信

系统控制要求如下：

1）当 2 号站（MPI 地址）的 I0.0 为 ON 时，3 号站（MPI 地址）的 Q0.0 输出为 ON 并保持；当 2 号站的 I0.1 为 ON 时，3 号站（MPI 地址）的 Q0.0 输出为 OFF 并保持。

2）当 3 号站的 I0.0 为 ON 时，2 号站的 Q0.0 输出为 ON 并保持；当 3 号站的 I0.1 为 ON 时，2 号站的 Q0.0 输出为 OFF 并保持。

（1）系统配置

系统配置如图 9-2 所示。CPU315-2PN/DP 采用 DP 接口通信，CPU314C-2DP 采用 MPI 接口通信，连接电缆采用 PROFIBUS-DP 电缆，并将电缆终端电阻打到“ON”状态。

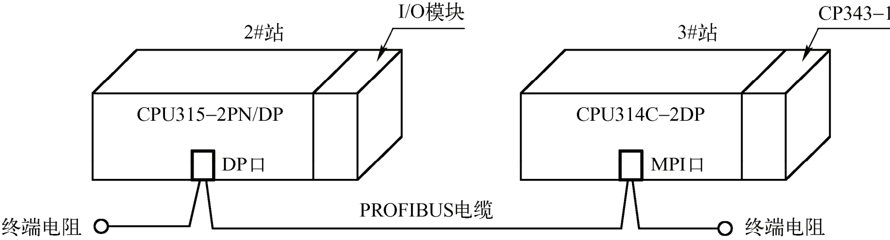


图 9-2　系统配置

（2）硬件组态

建立项目“MPI_TEST”，分别添加两台 PLC 作 MPI 网络 2 号站、3 号站，并组态硬件。组态硬件时，需要设置 MPI 地址及传输速率，如图 9-3 所示，MPI 地址设为 2，传输率可以选择默认速率 187.5kbit/s；同理，可以设置 3 号站。

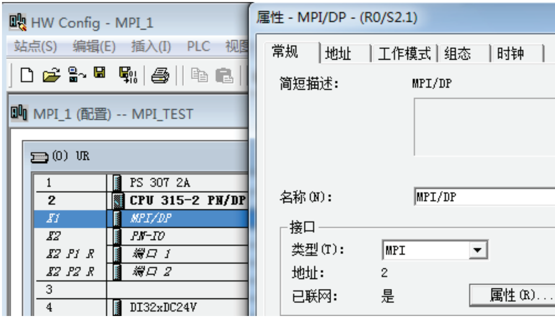


图 9-3　2 号站的硬件组态

项目硬件组态完毕后，回到管理界面，项目结构如图 9-4 所示。

（3）全局变量的建立

鼠标右键单击图 9-4 所示的“MPI(1)”网络，弹出图 9-5 界面，选中“定义全局数据”，弹出图 9-6 界面。

在图 9-6 界面中，选中空白的第 1 列，右键单击选中 CPU，弹出图 9-7 界面，选择需要通信的第 1 个 CPU 单击“确认”按钮，同样方法添加第 2 列 PLC，添加完成界面如图 9-8 所示。

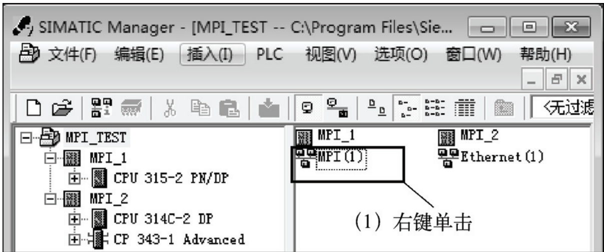


图 9-4 项目结构

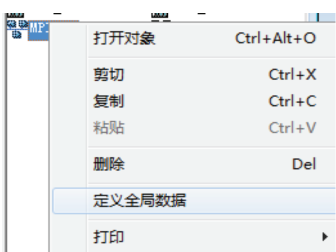


图 9-5 选择“定义全局数据”



图 9-6 设置全局变量界面

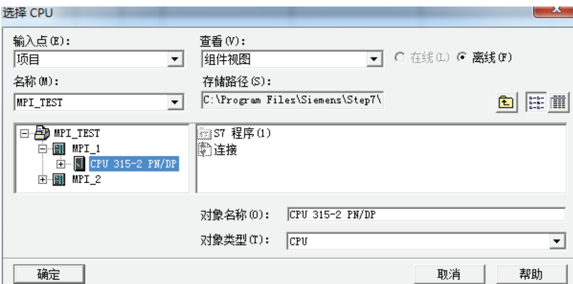


图 9-7 选择 CPU

添加通信站点完成后，添加通信变量，如图 9-9 所示，右键单击空白处，弹出下拉菜单，选择通信变量属性：发送或接收信息。分别添加两个站的通信变量，设置属性如图 9-10 所示，例如，MB0：1，“MB0”为发送信息的首地址，“1”为长度，表示 1 个字节长度。

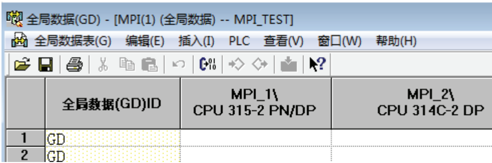


图 9-8 选择 CPU 完成界面

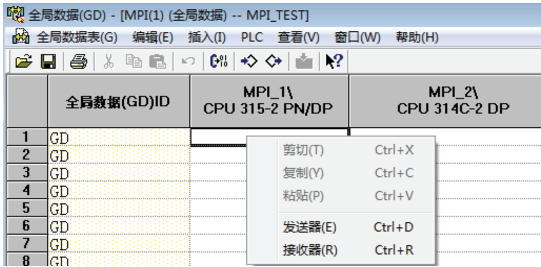


图 9-9 添加全局变量

	全局数据(GD)ID	MPI_1\ CPU 315-2 PN/DP	MPI_2\ CPU 314C-2 DP
1	GD	>MB0:1	QB0:1
2	GD	QB0:1	>MB0:1
3	GD	接收信息	发送信息

图 9-10 通信变量及内容

在输入数据时需要注意：无论是输入要发送的全局数据地址还是接收的全局数据地址都只能是绝对地址，不能是符号地址；地址区可以是 DB、M、I、Q，S7-300 CPU 地址区长度最大为 22B；发送区和接收区长度必须一致，即同一行中各个单元的字节数应相同；在每一行中定义一个且只能定义一个 CPU 作为数据的发送方，而接收方可以有多个 CPU。

完成通信变量设置后，在图 9-11 中单击“编译”快捷键，会出现 GD ID 编号，其格式为 GDA.B.C。A 为全局数据包的循环数，每一个循环数表示和一个 CPU 通信，循环数与 CPU 有关，对于 S7-300，最多为 4 个 CPU，每两个 CPU 通信的发送与接收是一个循环；B 为全局数据包的个数，表示一个循环有几个循环全局数据包，如果两个站相互通信，一个循环有两个数据包；C 为一个数据包的数据区数，一个数据包最大为 22B（见表 9-1），在本例中，如果 2 号站给 3 号站发送两组数据，则数据区数就是 2。

全局数据(GD) - [MPI(1) (全局数据) -- MPI_TEST]　编译

全局数据表(G)　编辑(E)　插入(I)　PLC　查看(V)　窗口(W)　帮助(H)

	全局数据(GD)ID	MPI_1\ CPU 315-2 PN/DP	MPI_2\ CPU 314C-2 DP
1	GD 1.1.1	>MB0	QB0
2	GD 1.2.1	QB0	>MB0

图 9-11　全局变量

（4）功能的实现

2 号站/3 号站主程序如图 9-12 所示，M0.0 为每个站的受控信号，根据图 9-11 所示的变量对应关系，该信号被发送到通信的对方站，并被 Q0.0 接收，从而控制对方站的 Q0.0 状态。图 9-12a 为按下 3 号站起动按钮时 2 号站 OB1 程序在线监控情况，图 9-12b 为按下 2 号站起动按钮时 3 号站 OB1 程序在线监控情况。

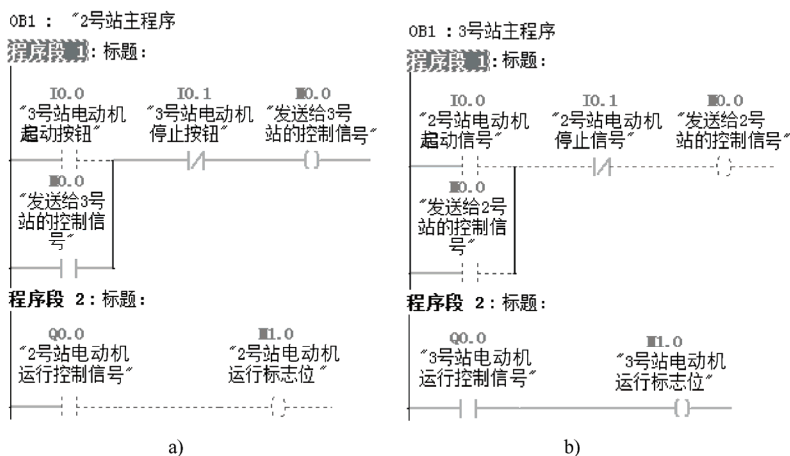


图 9-12　2 号站/3 号站主程序

a) 3 号站起动 2 号站电动机　b) 3 号站电动机运行状态

9.1.3　调用通信功能块通信

全局数据包通信的组态必须在同一项目下完成，因此缺乏灵活性，可以通过调用系统功能 SFC65~SFC69 来实现 MPI 的通信，这种通信方式适合于 S7-400/300/200 PLC 之间通信，且不需要组态。

一些旧型号的 S7-300/400 PLC 由于不含有 SFC65~SFC69，所以不能用这种通信方式，只

能用全局数据包的通信方式。判断一个 CPU 是否含有通信的 SFC，可以在联机的情况下在线查看所有的程序块，看系统功能是否包含 SFC65~SFC69。

通过调用 SFC 功能实现通信可分为两种方式：双边通信方式和单边通信方式。

双边通信方式要求通信双方都需要调用通信块，一方调用发送块发送数据，另一方调用接收块来接收数据，适用于 S7-300/400 之间通信，发送块是 SFC65（X_SEND），接收块是 SFC66（X_RCV）。

与双边通信方式不同，单边通信方式只需要在一方编写通信程序，也就是客户机与服务器的访问模式，例如 S7-200 PLC 作服务器，无须编写程序，S7-300 PLC 作客户端，在程序中调用 SFC67/SFC68，来实现 S7-200 与 S7-300 PLC 之间的 MPI 通信。

【例 9-1】 采用双边通信方式实现 S7-300 PLC 之间的通信，控制要求同 9.1.2 节。

1．项目结构

在 2 号站（3 号站）中调用 SFC65，如果扫描时间太短，发送频率太快，对方没有响应，将加重 CPU 负荷，可以选择在 OB35 中调用发送块，发送任务间隔设为 100ms。在 CPU 硬件组态中，2 号站 OB35 时间设置如图 9-13 所示，同理可设置 3 号站 OB35 循环中断时间。

2 号站项目结构如图 9-14 所示。同理可建立 3 号站项目结构。

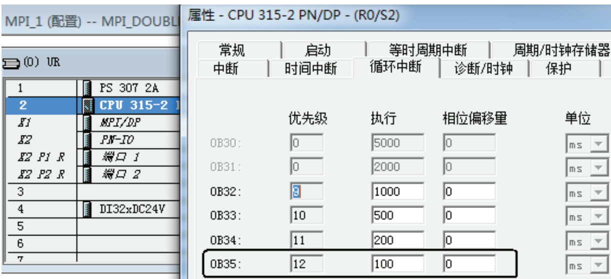


图 9-13 OB35 时间设置

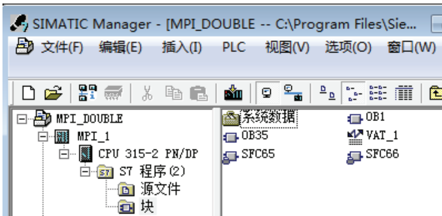


图 9-14 2 号站项目结构

2．SFC65/ SFC66 的应用

SFC65/SFC66 进入路径：在 OB35 程序编辑界面，单击“视图”→“Libraries”→“Standard Library”→“System Function Blocks”，选择 SFC65/SFC66，梯形图中出现如图 9-15 所示符号，参数说明见表 9-2。

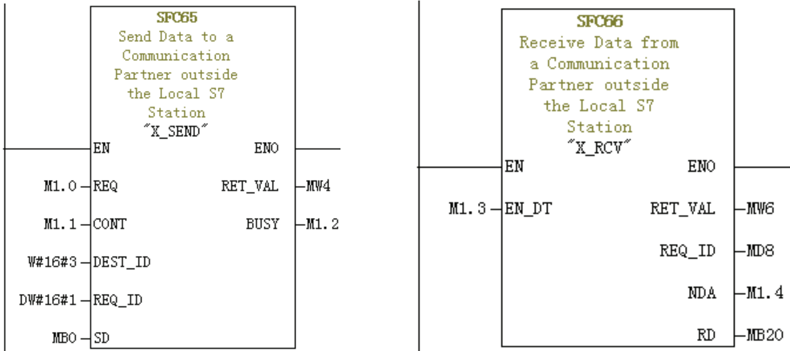


图 9-15 SFC65/ SFC 66 梯形图符号

表 9-2a　SFC65 参数说明

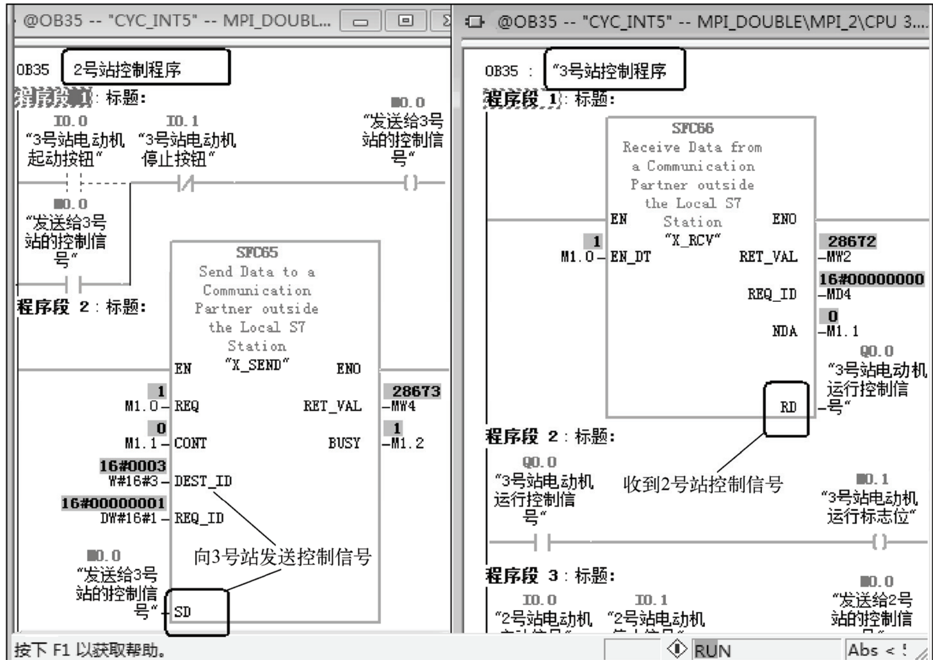
参 数 名 称	数 据 类 型	参 数 含 义
REQ	BOOL	发送请求，该参数为 1 时发送
CONT	BOOL	为 1 时表示发送数据是连续的一个整体
DEST_ID	WORD	接收方（通信对方 PLC）的 MPI 地址
REQ_ID	WORD	任务标识符
SD	ANY	本地 PLC 的数据发送区
RET_VAL	WORD	故障信号
BUSY	BOOL	通信进程："1"表示正在发送，"0"表示发送完成

表 9-2b　SFC66 参数说明

参 数 名 称	数 据 类 型	参 数 含 义
EN_DT	BOOL	接收使能
RET_VAL	WORD	错误代码：W#16#7000 表示无错
REQ_ID	DWORD	接收数据包的标识符
NDA	BOOL	通信进程："1"表示有新的数据包，"0"表示没有新的数据包
RD	ANY	本地 PLC 的数据接收区

3．程序设计与调试

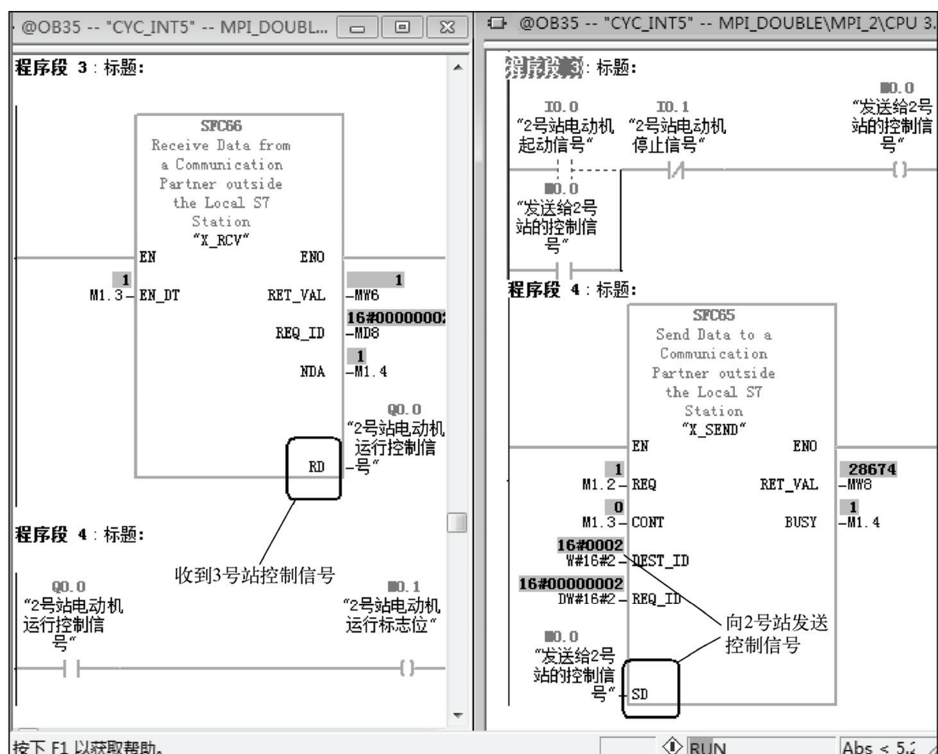
每个 MPI 站程序中都包括两部分程序，一部分是生成对方需要的信息并通过 SFC65 把数据发送出去，另一部分是接收对方发送的数据并应用该数据。本例中将满足控制要求的程序放在 OB35 中，程序设计和功能实现如图 9-16 所示。



a)

图 9-16　程序设计和功能实现

a) 2 号站发送/3 号站接收数据



b)

图 9-16 程序设计和功能实现（续）

b) 3 号站发送/2 号站接收数据

9.2 PROFIBUS 通信协议及其应用

9.2.1 PROFIBUS 通信协议

1．概述

PROFIBUS 是 Process Field Bus（过程现场总线）的简称，是 1987 年由 SIEMENS、ABB 等 13 家公司和 5 家研究机构按照 ISO/OSI 参考模型联合开发并制定的一种现场总线标准，符合德国国家标准 DIN19245 和欧洲标准 EN50170，也是国际标准 IEC61158 的组成部分 Type3。2001 年成为中国机械行业推荐标准 JB/T10308-2001，2006 年成为中国第一个工业通信领域现场总线技术国家标准 GB/T20540-2006。

PROFIBUS 产品在欧洲市场占有份额居首位，占有率超过 40%；在世界市场份额超过 20%，居于所有现场总线之首。根据 PNO（PROFIBUS Nutzer Organisation，PROFIBUS 用户组织）最新发布的数字，截止 2008 年，PROFIBUS 全球安装节点总数已超过了 2800 万，预测 2012 年将突破 5000 万大关。目前世界上许多自动化设备制造商都为他们生产的设备提供了 PROFIBUS 接口，PROFIBUS 现场总线已广泛应用于加工制造、过程控制、楼宇自动化以及交通电力等应用领域。

PROFIBUS 是一种应用广泛的、开放的、不依赖于设备生产商的现场总线标准，适合于快

速、时间要求严格的应用和复杂的通信任务。系统由主站和从站组成，主站能够控制总线、决定总线的数据通信，当主站得到总线控制权时，没有外界请求也可以主动发送信息；从站没有控制总线的权力，但可以对接收到的信息给予确认或者当主站发出请求时向主站回应信息。

2．PROFIBUS 总线的分类

PROFIBUS 现场总线根据应用的特点和用户不同的需要，可分为 PROFIBUS-DP、PROFIBUS-FMS 和 PROFIBUS-PA 三个互相兼容版本的通信协议，其中 PROFIBUS-DP 的应用最广。

1）PROFIBUS-DP（Distributed Periphery，分布 I/O 系统）是一种经过优化的高速、廉价的通信连接，主要用于自动化系统中单元级和现场级通信。可取代价格昂贵的 4~20mA 或 DC24V 并行信号线，实现自控系统和分散外围 I/O 设备及智能现场仪表之间的高速数据通信，传输速率达 12Mbit/s，一般构成单主站系统，适合于加工自动化领域的应用。

2）PROFIBUS-FMS（Fieldbus Message Specification，现场总线信息规范）可以用于车间级监控网络，主要解决车间级通用性通信任务，提供大量的通信服务，完成中等速度的循环和非循环通信任务，多用于纺织工业、楼宇自动化、电气传动、传感器和执行器、PLC 等自动化控制，一般构成实时多主站网络系统，是一种令牌结构、实时的多主网络。对于 FMS 而言，它考虑的主要是系统功能而不是响应时间，主要用于大范围的、复杂的通信系统。

3）PROFIBUS-PA（Process Automation，过程自动化）是应用于工业现场控制的过程自动化，遵从 IEC1158-2 标准，提供标准的本质安全的传输技术，一般用于安全性要求较高的场合及由总线供电的站点。

采用 PROFIBUS 标准系统，不同制造商所生产的设备不需对其接口进行特别调整就可实现通信，PROFIBUS 可用于高速并对时间苛求的数据传输，也可用于大范围的复杂通信场合，如图 9-17 所示。其中现场级由现场智能设备、现场智能仪表、DP/PA 耦合器等设备构成，涉及 PROFIBUS-DP 和 PROFIBUS-PA 两种通信协议。

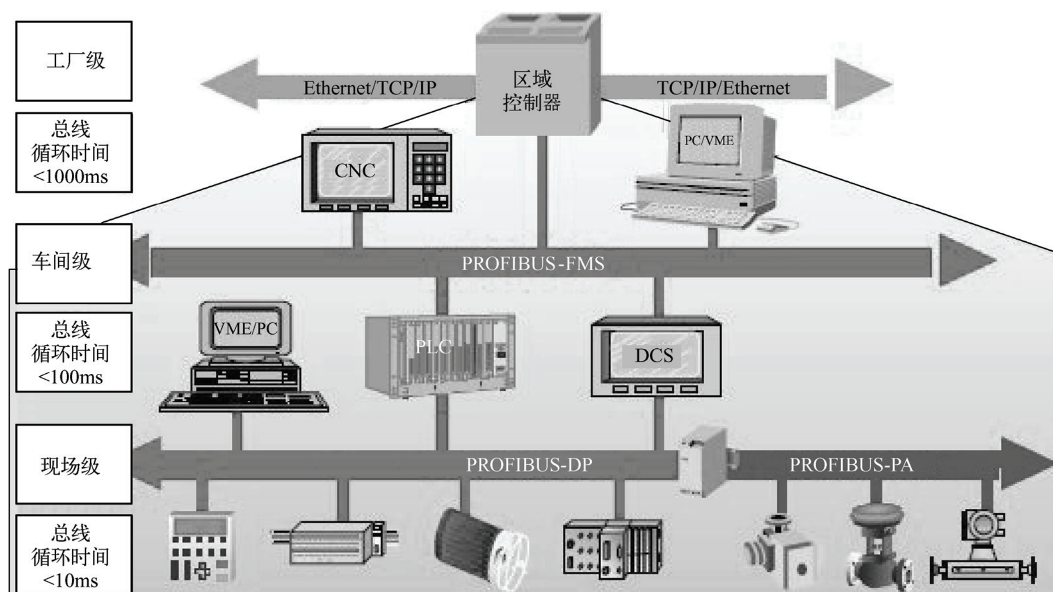


图 9-17　PROFIBUS 应用范围示意图

3．PROFIBUS 的协议结构

PROFIBUS 是根据 ISO7498 国际标准，以 OSI 参考模型为基础的。第 1 层为物理层，用来定义物理传输特性；第 2 层为数据链路层，用来解决两个相邻节点之间的通信问题；第 3~6 层 PROFIBUS 未使用；第 7 层为应用层，用来定义应用功能。协议结构示意图如 9-18 所示。

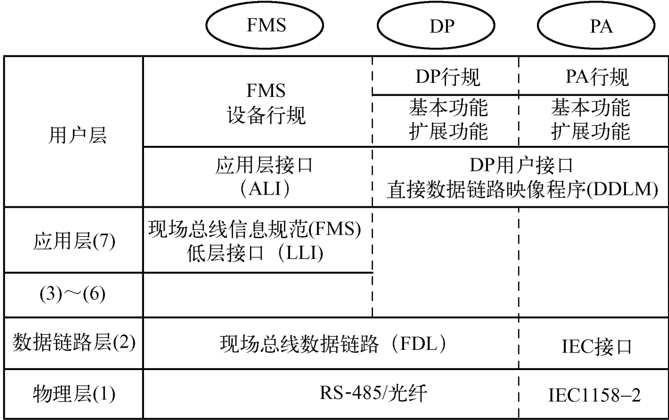


图 9-18　PROFIBUS 协议结构示意图

（1）PROFIBUS-DP

PROFIBUS-DP 定义了第 1、2 层和用户接口层。直接数据链路映像程序（DDLM）提供对第 2 层的访问，第 3~7 层未加描述，这种简化的协议结构保证了数据传输的快速性和有效性。该模型提供了 RS-485 传输技术和光纤传输技术；详细说明了各种不同 PROFIBUS-DP 设备的设备行为；定义了用户、系统以及不同设备可以调用的应用功能。特别适合可编程控制器与现场分散的 I/O 设备之间的快速通信。

（2）PROFIBUS-FMS

PROFIBUS-FMS 定义了第 1、2 层和第 7 层，第 3~6 层没有定义。第 7 层由现场总线信息规范（Fieldbus Message Specification，FMS）和低层接口（Lower Layer Interface，LLI）组成。PROFIBUS-FMS 包含应用协议，并向用户提供强有力的通信服务；LLI 协调不同的通信关系并向 FMS 提供不依赖于设备的对第 2 层访问的接口。

FMS 处理单元级的数据通信，功能强大，应用广泛，但近年来随着以工业以太网和 TCP/IP 协议为基础的 PROFINET 的应用和普及，其功能逐渐被取代。

由于 PROFIBUS-FMS 和 PROFIBUS-DP 使用相同的传输技术和总线存取协议，因此它们可以在同一根电缆上同时运行。

（3）PROFIBUS-PA

PROFIBUS-PA 采用扩展的 PROFIBUS-DP 协议进行数据传输，另外它还使用了描述现场设备行为的 PA 规范。根据 IEC1158-2 标准，这种传输技术可确保其本质安全，并使现场设备通过总线供电。使用 DP/PA 耦合器和 DP/PA LINK 连接器，PROFIBUS-PA 设备能很方便地集成到 PROFIBUS-DP 网络上。

PROFIBUS-PA 是为满足过程自动化工程中高速、可靠的通信要求而特别设计的。用 PROFIBUS-PA 可以把传感器和执行器连接到现场总线上，即使在防爆区域的传感器和执行器

也是如此。

4．总线存取协议

PROFIBUS-DP、PROFIBUS-FMS 和 PROFIBUS-PA 均使用一致的总线存取协议，通过 OSI 参考模型的第 2 层数据链路层来实现。介质存取控制（Medium Access Control，MAC）必须确保在任何时刻只能由一个站点发送数据。PROFIBUS 协议的设计要满足介质控制的两个基本要求：其一，同一级的 PLC 或主站之间的通信必须使每一个主站在确定的时间范围内能获得足够的机会来处理它自己的通信任务；其二，主站和从站之间应尽可能快速而又简单地完成数据的实时传输。为此，PROFIBUS 使用混合的总线存取控制机制来实现上述目标，包括用于主站之间通信的令牌传送方式和用于主站与从站之间通信的主从方式。

当一个主站获得了令牌，它就可以拥有主从站通信的总线控制权，而且此地址在整个总线上必须是唯一的。在一个总线内，最大可使用的站地址范围是在 0~126 之间，也就是说，一个总线系统最多可以有 127 个节点。

这种总线存取控制方式可以有以下 3 种系统配置。

1）纯主-主系统（令牌传送方式）。

2）纯主-从系统（主从方式）。

3）两种方式的组合。

PROFIBUS 的总线存取控制符合欧洲 EN50170 V.2 中规定的令牌总线程序和主-从程序，与所使用的传输介质无关。

（1）令牌总线通信过程

连接到 PROFIBUS 网络的主站按它的总线地址的升序组成一个逻辑令牌环，如图 9-19 所示。在逻辑令牌环中控制令牌按照事先给定的顺序从一个站传递到下一个站，令牌提供控制总线的权力，并用特殊的令牌帧在主站点间进行传递。具有最高站地址（Highest Address Station，HAS）的主站点例外，它只把令牌传递给具有最低总线地址的主站点，以此使逻辑令牌环闭合。令牌环调度要保证每个主站有足够的时间来完成它的通信任务。令牌经过所有主站点轮转一次所需时间叫做实际令牌循环时间 TRR，每一次令牌交换都会计算产生一个新的 TRR；用目标令牌时间 TTR 来规定现场总线系统中令牌轮转一次所允许的最大时间，这个时间是可以调整的；一个主站在获得令牌后，就是通过计算 TTR-TRR 来确定自己持有令牌的时间 TTH。

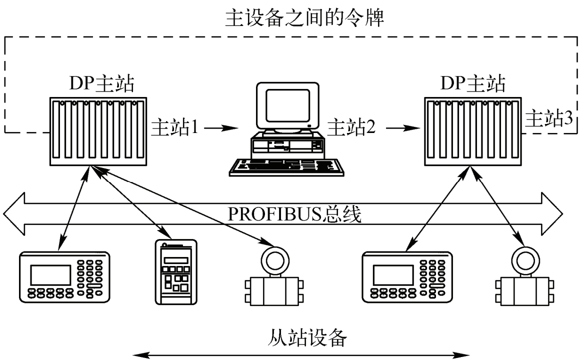


图 9-19　PROFIBUS 系统的多主结构示意图

在总线初始化和启动阶段，MAC 通过辨认主站点来建立令牌环。为了管理控制令牌，MAC 程序首先自动地判定总线上所有主站点的地址，并将这些节点及它们的节点地址都记录在活动主站表（List of Active Master Stations，LAS）中。对于令牌管理而言，有两个地址概念特别重要：一个是前一站（Previous Station，PS）节点的地址，即下一站是从此站接收到令牌的；另一个是下一站（Next Station，NS）节点的地址，即令牌传递给此站。

在运行期间，为了从令牌环中去掉有故障的主站点或增加新的主站点到令牌环中而不影响总线上的数据通信，也需要用到 LAS。若一个主站从 LAS 中自己的 PS 站收到令牌，则保留令牌并使用总线；若主站收到的令牌帧不是 PS 站发出的，将认为是一个错误而不接收令牌；如果此令牌帧被再次收到，该主站将认为令牌环已修改，接收令牌并修改自己的 LAS。

（2）主/从通信过程

一个网络中有若干个从站，而它的逻辑令牌环只含一个主站，这样的网络就称为纯主-从系统，如图 9-20 示，此系统不存在令牌的传递。主/从通信允许主站控制它自己所控制的从站，使得从站作出相应的响应；主站要与每一个从站建立一条数据链路；主站可以发送信息给从站或者获取从站信息。

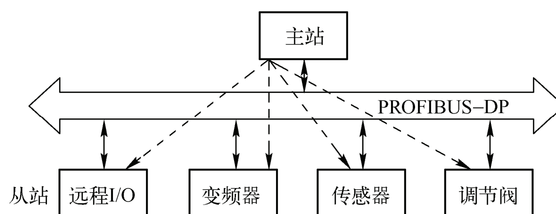


图 9-20　PROFIBUS 主从通信过程

主/从（Master/Slave，MS）通信方式是 PROFIBUS-DP 主站与智能从站之间的数据交换方式，可以由 PLC 的操作系统周期性地自动完成，不需要用户程序进行控制。但用户必须对主站和智能从站之间的通信连接和数据交换区进行配置。

在分布式 PLC 系统中，PLC 可以被设为主站，通过 PROFIBUS-DP 总线连接分布式 I/O 从站，如 ET200B 紧凑型 DP 从站、ET200M 模块式 DP 从站等，这些从站实质上只是带有 PROFIBUS-DP 通信处理器的 I/O 模块，称为标准从站或普通从站。标准从站的 I/O 被直接并入 DP 主站的 I/O 地址区，使用时可以像主站本身的 I/O 模块一样直接访问标准从站的输入/输出。

但是，对于有些控制系统，可能带有多台 PLC 或其他带有 CPU、存储器等部件的独立控制设备，以实现不同子任务的独立和有效处理。同时，整个控制系统为了实现分散控制、集中管理，将这些设备都挂接在 PROFIBUS-DP 网络上。这些独立的控制设备在 DP 网络中被称为“智能从站”。

智能从站本身具有独立的 I/O 地址区，这些地址可能会与主站的 I/O 地址相同，因此，DP 主站不能直接访问智能从站的输入/输出，而是需要建立输入/输出地址的传输空间，并由智能从站的 CPU 负责处理地址转换工作。在通信配置时需要注意，被指定用于主站和从站之间交换数据的输入/输出区，不能占据主站安装有 I/O 模块的物理地址区。

（3）两种方式的组合

一个 DP 系统可能是多主结构，这意味着一条总线上连接几个主站节点，主站间采用逻辑令牌环、主从站间采用主/从通信的方式传输。

令牌传递程序保证每个主站在一个确切规定的时间内得到总线存取权；主站得到总线控制权时，可与从站进行主/从通信，对从站进行分时轮询传输信息。

在图 9-19 中，总线系统由 3 个主站和 5 个从站构成。3 个主站之间构成令牌逻辑环：主站 1→主站 2→主站 3→主站 1，当其中一个主站得到令牌报文后，该主站就在一定时间内执行主站工作；在这段时间内，它可依照主-从关系通信表与所有从站通信，也可依照主-主关系通信表与所有主站通信。

9.2.2 PROFIBUS 的传输技术

现场总线系统的应用在较大程度上取决于采用哪种传输技术，而且既要考虑传输的拓扑结构、传输速率、传输距离和传输的可靠性等通用要求，还要考虑成本低廉、使用方便等因素；在过程自动化的应用中，为了满足本质安全的要求，数据和电源还必须在同一根传输媒介上传输，因此单一的传输技术不可能满足以上所有要求。

在通信模型中，物理层直接和传输介质相连，规定了线路传输介质、物理连接的类型、电气、功能等特性，提供了三种数据传输类型，分别是用于 DP 和 FMS 的 RS-485 传输技术、用于 DP 和 FMS 的光纤传输技术以及用于 PA 的 IEC1158-2 传输技术。

1．RS-485 传输技术

RS-485 是一种简单的、低成本的传输技术，其传输过程是建立在半双工、异步、无间隙同步化的基础上，数据的发送采用 NRZ 编码，这种传输技术通常称为 H2。网络连接情况如图 9-21 所示，具有以下特点：

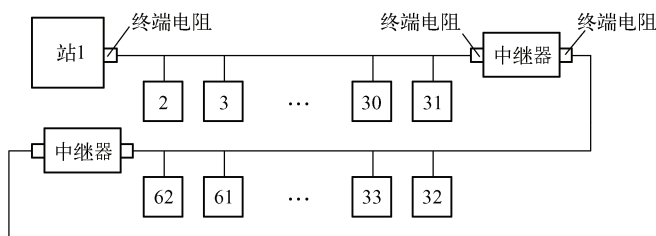


图 9-21　使用中继器连接各总线段

1）网络拓扑：所有设备都连接在总线结构中，每个总线段的开头和结尾均有一个终端电阻。为确保操作运行不发生误差，两个总线终端电阻必须要有电源。

2）传输介质：双绞屏蔽电缆，也可取消屏蔽，取决于电磁干扰环境（Electro Magnetic Compatibility，EMC，电磁兼容性）的条件。

3）站点数：每个总线段最多可以连接 32 个站，如果站数超过 32 个或需要扩大网络区域，则需要使用中继器来连接各个总线段。使用中继器时最多可用到 127 个站，串联的中继器一般不超过 3 个。

4）插头连接：采用 9 针、D 型插头，插座被安装在设备上，9 针 D 型连接器的针脚分配见表 9-3。

表 9-3　9 针 D 型连接器的针脚分配

针　脚　号	信　　号	信 号 含 义
1	Shield	屏蔽/保护地
2	M24	24V 输出电压的地
3	RxD/TxD-P	接收数据/发送数据（正）
4	CNTR-P	中继器控制信号（方向控制）
5	DGND	数据基准电压
6	VP	供电电压
7	P24	输出电压 24V
8	RxD/TxD-N	接收数据/发送数据（负）
9	CNTR-N	中继器控制信号（方向控制）

5）传输速率：可以在 9.6 kbit/s~12Mbit/s 之间选择各种传输速率。

6）传输距离：总线的最大传输距离取决于传输速率，范围为 100~1000m。对应关系见表 9-4，若有中继器距离可延长到 10km。

表 9-4　传输速率所对应的最大允许段长度

波特率/kbit/s	9.6	19.2	93.75	187.5	500	1500	12000
段长度/m	1200	1200	1200	1000	400	200	100

RS-485 传输的数据位格式如图 9-22 所示。

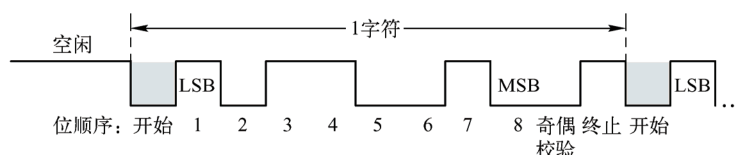


图 9-22　数据传输格式

数据的发送采用 NRZ 编码方式，1 个字符帧为 11 位。每个八位二进数字节按最小的有效位（Least Significant Bit，LSB）先发送，最高的有效位（Most Significant Bit，MSB）最后被发送的顺序传输；每个八位二进数都补充三位即开始、奇偶校验和终止位。用高电位表示“1”，零电位表示“0”；表示“1”的高电位脉冲中途不归零。

站与站数据线的连接方式如图 9-23a 所示，两根 PROFIBUS 数据线也常被称为 A 线和 B 线，A 线对应于 RxD/TxD-N 信号，B 线对应于 RxD/TxD-P 信号。总线终端电阻的连接结构如图 9-23b 所示，包括一个相对于 DGND 数据参考电势的下拉电阻和一个相对于输入正电压 VP 的上拉电阻；当总线上没有发送数据时，即在两个报文之间总线处于空闲状态，这两个电阻也能确保在总线上有一个确定的空闲电位。

在数据传输期间，二进制“1”对应于 RXD/TXD-P 线上的正电位，而在 RXD/TXD-N 线上则相反；各报文间的空闲状态对应于二进制“1”信号，如图 9-24 所示。

RS-485 总线的连接结构如图 9-25 所示。根据 EIA RS-485 标准，在数据线 A 和 B 的两端均要连接总线终端器，几乎在所有标准的 PROFIBUS 总线连接器上都组合了所需要的总线终

端器，可以由跳接器或开关来启动。

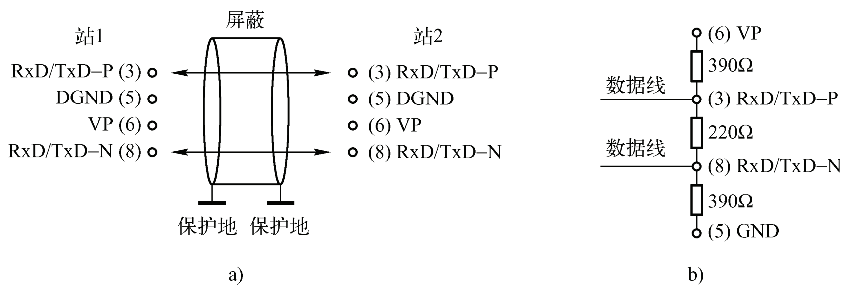


图 9-23 总线的连接

a) 数据线的连接 b) A 型电缆总线终端电阻的连接

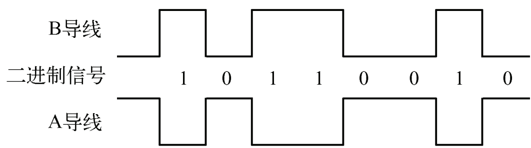


图 9-24 A/B 导线的电位

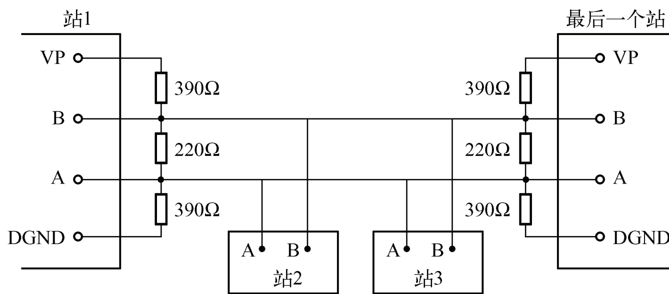


图 9-25 RS-485 总线结构

2．DP/FMS 的光纤传输技术

在电磁干扰很大的环境或需要覆盖很远的传输距离的网络应用中，可使用光纤传输技术。光纤是一种采用玻璃作为波导，以光的形式将信息从一端传送到另一端的技术。光纤电缆对电磁干扰不敏感并能保证总线上站与站的电气隔离，允许 PROFIBUS 系统站之间的距离最大为 15km。

现在的低损耗玻璃光纤相对于早期发展的传输介质，几乎不受带宽限制且传输距离远、衰减小；点到点的光学传输系统由三个基本部分构成：产生光信号的光发送机、携带光信号的光缆和接收光信号的光接收机。

许多厂商提供专用总线插头可将 RS-485 信号转换成光纤信号，或者将光纤信号转换成 RS-485 信号，这使得在同一系统中，可同时使用 RS-485 传输技术和光纤传输技术。为了使光纤导体与总线站连接，可以采用 OLM（Optical Link Module）技术、OLP（Optical Link Plug）技术和集成的光缆连接等技术。

3．PA 的 IEC1158-2 传输技术

IEC1158-2 传输技术能满足化工、石化等工业对环境的要求，可保证本质安全性和现场设备通过总线供电。这是一种位同步协议，可进行无电流的连续传输，通常称为 H1，采用曼彻斯特编码，能进行本质及非本质安全操作；每段只有一个电源作为供电装置，当站收发信息时，不向总线供电。具体特性如下：

1）数据传输：数字式、位同步、曼彻斯特编码。

2）传输速率：通信速率为 31.25kbit/s，与系统结构和总线长度无关。

3）数据可靠性：前同步信号，采用起始和终止限定符避免误差。

4）传输介质：可以采用屏蔽双绞线电缆，也可以采用非屏蔽式双绞线电缆。

5）远程电源供电：为可选附件，可通过数据总线供电。

6）防爆型：能进行本征及非本征安全操作。

7）站点数：每段最多有 32 个站；使用中继器最多可到 127 个站。

8）拓扑结构：采用总线型、树型或混合型网络拓扑结构。

PA 总线电缆的终端各有一个无源 RC 线路终端器，如图 9-26 所示。一个 PA 总线上最多可连接 32 个站点，总线的最大长度取决于电源、传输介质的类型和总线站点的电流消耗。

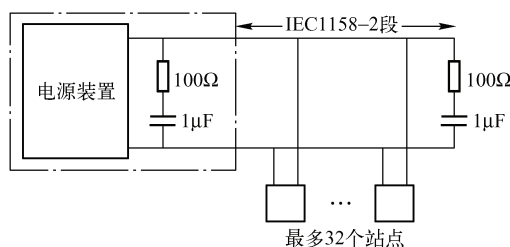


图 9-26　PA 总线的结构

9.2.3　PROFIBUS 控制系统的配置

PROFIBUS 网络的硬件由主站、从站、网络部件和网络工具等组成。其中网络部件包括通信介质（例如电缆、光缆等）、总线连接器（例如中继器、RS-485 总线连接器等），以及用于连接串行通信、以太网、AS-i（Actuator Sensor Interface，传感器/执行器接口）、EIB（Electrical Installation Bus，电气安装总线）等网络系统的网络转换器；网络工具包括 PROFIBUS 网络配置、诊断的软件与硬件，用于网络的安装与调试。

1．现场设备的分类

根据现场设备是否具有 PROFIBUS 接口分为：

1）现场设备不具备 PROFIBUS 接口。采用分布式 I/O 作为现场设备与总线连接的接口，这种形式在应用现场总线技术初期容易推广。

2）部分现场设备具有 PROFIBUS 接口。这是目前现场总线系统中普遍存在的形式，系统可以采用具有 PROFIBUS 接口的现场设备与分布式 I/O 混合使用的办法。

3）现场设备都有 PROFIBUS 接口。这是一种理想状况，可以使现场设备直接通过

PROFIBUS 接口接入现场总线系统，从而形成完全的分布式结构。但采用这种方案，设备和系统成本都会增高。

无论是旧系统改造还是新建项目，全部使用具备 PROFIBUS 接口的现场设备的场合可能不多，因此分布式 I/O 可作为通用的现场总线接口，是一种灵活的系统集成方案。

根据现场设备在控制系统中所起的作用不同可分类如下。

1）1 类主站（DPM1）。1 类主站（DPM1）是中央控制器，完成总线通信控制、管理及周期性数据访问。无论 PROFIBUS 的网络采用何种结构，1 类主站是系统必须的。比较典型的 DPM1 有 PLC、PC 以及支持主站功能的各种通信处理器模块等设备。

2）2 类主站（DPM2）。2 类主站（DPM2）完成非周期性数据访问，如数据读写、系统配置、故障诊断及管理组态数据等，它可以与 1 类主站进行通信，也可以与从站进行输入/输出数据的通信，并为从站分配新的地址。DPM2 主要是在工程设计、系统组态或操作设备时使用，比较典型的 DPM2 有编程设备、触摸屏及操作面板等设备。

3）从站。PROFIBUS 从站是对数据和控制信号进行输入/输出的设备。从站在主站的控制下，进行现场输入信号的采集与控制信号的输出。作从站的设备可以是 PLC 一类的控制器，也可以是不具有程序存储和程序执行功能的分散式 I/O 设备，还可以是像 SITRANS 现场仪表、MicroMaster 变频器等具有总线接口的智能现场设备。

2．PROFIBUS 控制系统的配置形式

根据实际应用需要，PROFIBUS 控制系统的配置可分为以下几种结构类型：

1）PLC 或其他控制器作 1 类主站，不设监控站。这种结构类型中，在调试阶段需要配置一台编程设备、PLC 或其他控制器完成总线通信管理、从站数据读写、从站远程参数化工作。

2）以 PLC 或其他控制器作 1 类主站，监控器通过串口与 PLC 连接。这种结构类型中，监控站不是 2 类主站，不能直接读取从站数据和完成远程参数化工作。监控站所需数据只能从 PLC 控制器中读取。

3）以 PLC 或其他控制器作 1 类主站，监控站连接在 PROFIBUS 总线上。这种结构类型中，监控站作为 2 类主站运行，可完成远程编程、参数设置及在线监控功能。

4）使用 PC+PROFIBUS 网卡作 1 类主站。这种结构类型中，PC 既作主站又作监控站，是个低成本的配置方案；但需要选用具有高可靠性、能长时间连续运行的工业级 PC，一旦 PC 发生故障将会导致整个系统瘫痪。

5）工业控制 PC+PROFIBUS 网卡+SoftPLC。SoftPLC 是一种软件产品，可以将通用型 PC 改造成一台由软件实现的 PLC；将 PLC 的编程功能、应用程序运行功能、操作员监控站的图形监控开发功能都集成到一台 PC 上，形成 PLC 与监控站于一体的控制器工作站。

3．混合系统的构建

（1）DP 与 PA 的连接

如图 9-27 所示，采用 DP/PA 耦合器可将两段不同协议的总线互连。DP 网段的波特率可变，而 PA 网段的波特率为固定值 31.25kbit/s。DP/PA 耦合器用于在 DP 和 PA 之间传递物理信号，适用于简单网络与运行时间要求不高的场合，分为两种类型：非本质安全型和本质安全型。

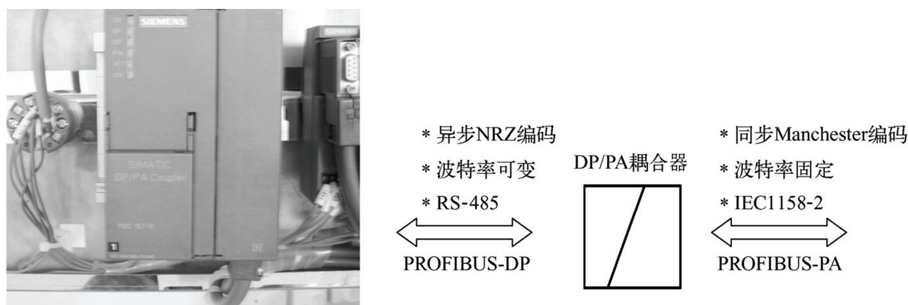


图 9-27　用耦合器转换协议

PA 现场设备还可以通过 DP/PA 链路设备连接到 DP 网络上。DP/PA 链路设备应用在大型网络时，依赖网络复杂程度和处理时间要求的不同，会有不止一个链路设备连接到 DP。DP/PA 链路设备既作为 DP 网段的从站又作为 PA 网段的主站，耦合网络上的所有数据通信；这意味着在不影响 DP 性能的情况下，DP/PA 链路设备将 DP 和 PA 结合起来，由于每个链路设备可以连接多台设备，而链路设备只占用 DP 的一个站地址，因此整个网络所能容纳的设备数量大大增加。

（2）DP 和 FMS 的混合

前已述及，PROFIBUS-DP 和 PROFIBUS-FMS 使用相同的传输技术和总线存取协议，不同的应用功能是通过第 2 层不同的服务访问点分开的。因此两个协议可以在同一条总线上进行混合操作，也可以在同一台设备上执行，如图 9-28 所示。

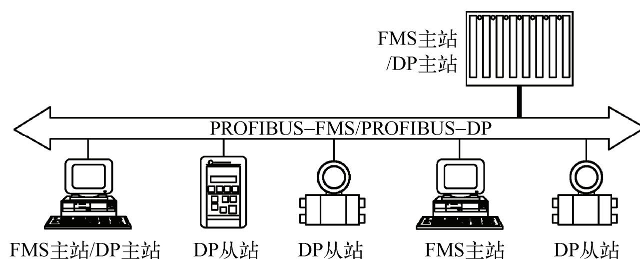


图 9-28　DP 与 FMS 的混合操作

9.2.4　GSD 文件

1．GSD 文件简介

GSD 文件用来描述 PROFIBUS-DP 设备的特性。为了将不同厂商生产的 PROFIBUS 产品集成在一起，生产厂商必须为其产品提供电子设备数据库文件，即 GSD 文件。该文件对 PROFIBUS 设备的特性诸如波特率、信息长度、诊断信息等进行了详细说明；标准化的 GSD 数据将通信扩大到操作员控制级；使用基于 GSD 的组态工具，可将不同厂商生产的设备集成在同一总线系统中，既操作简单又界面友好。

2．GSD 文件的组成

GSD 文件包含用于通信通用的和设备专用的规范，其文件结构可以分为以下三个部分：

1）一般规范。这部分包括制造商的信息、设备的名称、硬件和软件的版本状况、所支持

的传输速率、可能的监视时间间隔以及在总线连接器上的信号分配等。

2）与 DP 主站有关的规范。这部分包含所有与主站有关的参数，如最大可连接的从站个数、上装和下载选项等，这部分内容不能用于从站设备。

3）与 DP 从站有关的规范。这部分包括与从站有关的信息，如输入/输出通道的数量和类型、中断测试的规范、输入/输出数据一致性的信息等。

3．GSD 文件格式

GSD 是可读的 ASCII 文本文件，可以用任何一种 ASCII 编辑器编辑，如记事本、UltraEdit 等，也可使用 PROFIBUS 用户组织提供的编辑程序 GSDEdit 编辑。GSD 文件由若干行组成，每行都用一个关键字开头，包括关键字及参数（无符号数或字符串）两部分。借助于关键字，组态工具从 GSD 读取用于设备组态的设备标识、可调整的参数、相应的数据类型和所允许的界限值。GSD 文件中的关键字有些是强制性的，例如 VendorName，有些关键字是可选的，例如 SyncMode Supp。GSD 代替了传统的手册并在组态期间支持对输入错误及数据一致性的自动检查。

一个 GSD 文件的例子如下。

```
#PROFIBUS DP                        ；DP 设备的 GSD 文件均以此关键词存在
GSD Revision=1                      ；GSD 文件版本
VendorName="Meglev"                 ；设备制造商
Model Name="DP Slave"               ；从站模块
Revision="Version 01"               ；产品名称，产品版本
RevisionNumber=01                   ；产品版本号（可选）
IdentNumber=0x01                    ；产品识别号
 Protocol Ident=0                   ；协议类型（表示 DP）
 StationType=0                      ；站类型（0 表示从站）
 FMS Supp=0                         ；不支持 FMS，纯 DP 从站
 Hardware Realease="HW1.0"          ；硬件版本
Software Realease="SWl.0"           ；软件版本
9.6 supp=1                          ；支持 9.6kbit/s 波特率
19.2 supp=l                         ；支持 19.2kbit/s 波特率
MaxTsdr 9.6=60                      ；9.6kbit/s 时最大延迟时间
MaxTsdrl9.2=60                      ；19.2kbit/s 时最大延迟时间
RepeaterCtrl sig=0                  ；不提供 RTS 信号
24VPins=0                           ；不提供 24V 电压
Implementation Type="SPC3"          ；采用的解决方案
FreezeMode Supp=0                   ；不支持锁定模式
SyncMode Supp=0                     ；不支持同步模式
AutoBaud Supp=l                     ；支持自动波特率检测
Set SlaveAdd Supp=0                 ；不支持改变从站地址
Fail Safe=0                         ；故障安全模式类型
MaxUser PrmDataLen=0                ；最大用户参数数据长度（0~237）
Usel prmDataLen=0                   ；用户参数长度
Min Slave Imervall=22               ；最小从站响应循环间隔
Modular Station=l                   ；是否为模块站
MaxModule=l                         ；从站最大模块数
```

MaxInput Len=8　　　　　；最大输入数据长度
MaxOutput Len=8　　　　；最大输出数据长度
MaxData Len=16　　　　　；最大数据的长度（输入输出之和）
MaxDiagData Len=6　　　；最大诊断数据长度（6~244）Slave
：
EndModule

通过读 GSD 到组态程序中，用户可以获得最适合使用的设备专用通信特性。为了支持设备商，PROFIBUS 网站上有专用的 GSD 编辑/检查程序可供下载，便于用户创建和检查 GSD 文件，也有专用的 GSD 文件库供相关设备的用户下载使用。

9.2.5　S7-300 PLC 之间的 PROFIBUS-DP 通信

1．控制要求

采用 PROFIBUS-DP 通信方式，完成 S7-300 PLC 之间的信息交换和控制功能。控制要求如下：

1）主站控制从站电动机的运行和停止。

2）从站控制主站电动机的运行和停止。

3）按下起动按钮 3s 后电动机运行，同时电动机运行 5s 后停止，3s 后继续运行，如此循环。

2．控制系统硬件配置及结构图

根据系统控制要求，系统配置如下：CPU313C-2DP PLC 2 台；PC 1 台；PC/Adapter 编程电缆 1 根；PROFIBUS-DP 通信电缆 1 根。

CPU313C-2DP PLC 是紧凑型 CPU，适合安装在分布式结构中，集成的数字量 I/O 可直接与过程系统相连接，PROFIBUS-DP 主站/从站接口允许连接独立的 I/O 单元，可以用来建立高速、易用的分布式自动化系统。

系统结构如图 9-29 所示。

3．硬件组态

（1）新建项目

进入 SIMATIC Manager 界面，单击“文件”→“新建”菜单项，创建新项目“profibus”；然后单击“插入”→“站点”→“SIMATIC 300 Station”菜单项，插入两个 S7-300 站，分别命名 master 和 slave，如图 9-30 所示；然后分别组态主站和从站，在对两个 CPU 主/从站进行硬件组态时，原则上先组态从站。



图 9-29　系统结构图

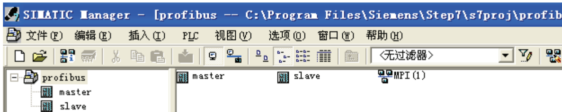


图 9-30　项目的建立

（2）从站组态

在 SIMATIC Manager 对话框内，双击“slave”图标，然后在右侧视窗内双击“slave”图标，进入硬件组态窗口。在工具栏内单击工具打开硬件目录，进行机架的建立，如图 9-31 所示。

机架的 1 号插槽为电源模块，本系统中 PLC 采用外部开关电源供电，因此 1 号插槽不用，2 号插槽为 CPU 模块，配置步骤如图 9-32 所示。由于 CPU313C-2DP PLC 集成了 16 个数字量输入点和 16 个数字量输出点，系统分别为其配置了两个字节的输入地址 IB124、IB125 和两个字节的输出地址 Q124、Q125。

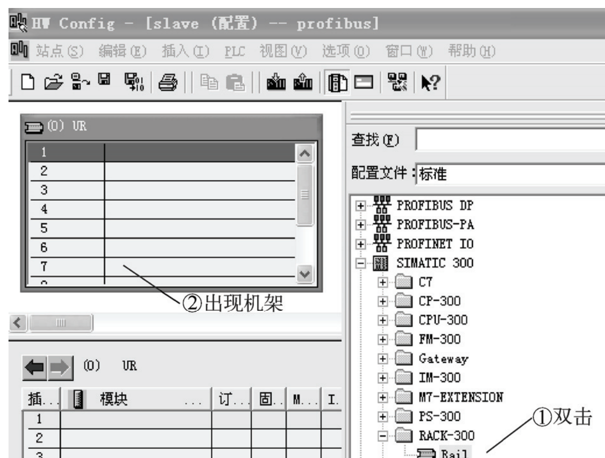


图 9-31　机架的建立

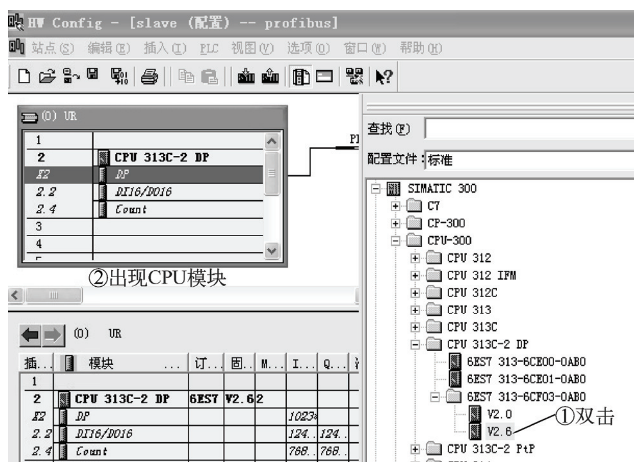


图 9-32　CPU 模块的配置

如果需要为输入输出分配其他地址，例如，IB0、IB1，则按照图 9-33 操作，将输入/输出的“系统默认”前面的勾取消，并将开始地址改为 0，修改后界面如图 9-34 所示。

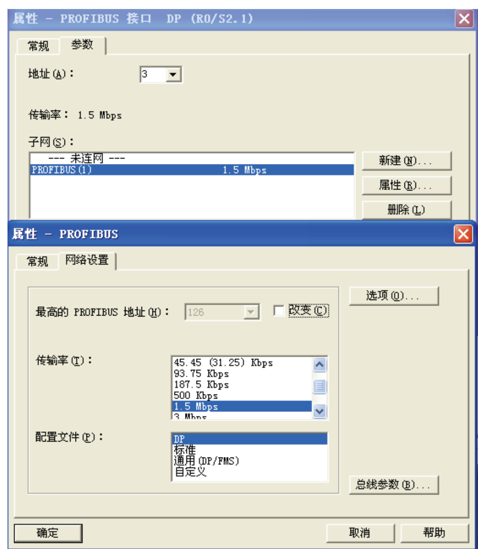
在往机架中插入 CPU 时会同时弹出 PROFIBUS 接口组态窗口，也可以在插入 CPU 后双击 DP 插槽，弹出“DP 属性”对话框，单击“属性”按钮进入 PROFIBUS 接口组态窗口，如图 9-35 所示。单击“新建”按钮新建 PROFIBUS 网络，分配 PROFIBUS 站地址，本系统将从站地址设为 3 号站；单击“属性”按钮组态网络属性，选择“网络设置”选项卡进行网络参数设置。

在从站组态时，需要设置从站的 DP 模式。鼠标右键单击 DP 插槽，选中“对象属性”弹

出属性对话框，然后选择“工作模式”选项卡，激活“DP 从站”模式，如图 9-36 所示。



图 9-33 PLC 输入/输出地址的修改



插...	模块	订货号	固...	M...	I...	Q...	注释
1	PS 307 5A	6ES7 307-1EA00-0AA0					
2	CPU 313C-2 DP	6ES7 313-6CF03-0AB0	V2.6	2			
X2	DP				1023*		
2.2	DI16/DO16				0...	0...1	
2.4	计数				768...	768...	
3							

图 9-34 PLC 输入/输出地址修改后界面

图 9-35 PROFIBUS 网络的建立

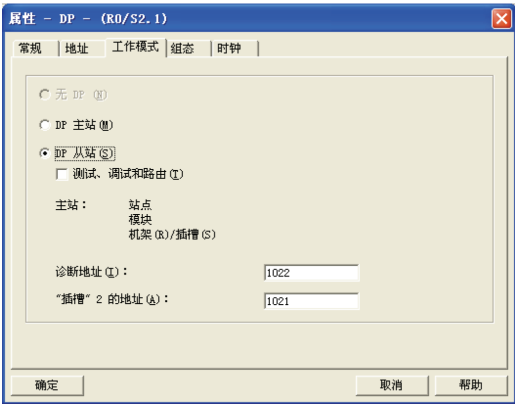


图 9-36 设置 DP 模式

在进入 DP 属性设置对话框中，选择“组态”选项卡，单击“新建”按钮新建一行通信接口区，如图 9-37 所示，设从站从主站接收一个字节的信息，字节地址为 IB2。

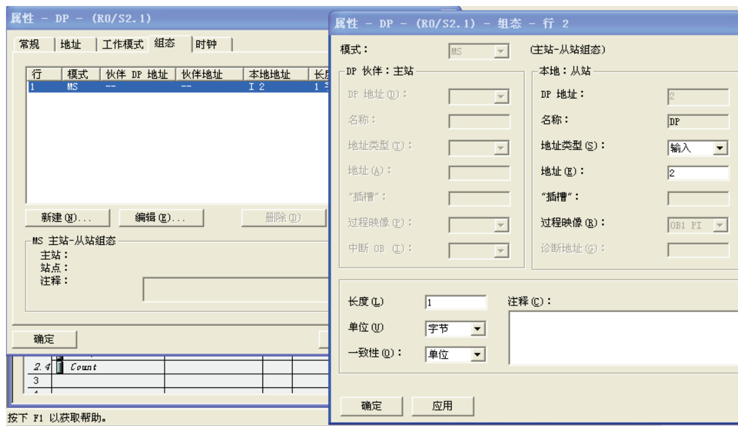


图 9-37　设置通信接口区

再在“属性-DP”页面中单击“新建”按钮，选择从站的地址类型为“输出”，地址为“3”，长度为“1”，建立 QB3 变量，以便从站给主站发送信息。从站“组态”完毕后的接口通信情况如图 9-38 所示。

当从站组态完成后单击“保存和编译”图标，对从站进行编译保存。若编译从站的时候出现“由于组态不一致而无法重新创建系统数据”的字样时，说明没有配置“属性-DP”对话框中“新建”通信接口。

（3）主站组态

完成从站的组态后，需要对主站进行组态。组态过程与从站基本相同，可参考从站的操作步骤。在完成基本硬件组态后，还需要对 DP 接口进行参数设置，本系统中主站地址设置为 2，并选择与从站相同的 PROFIBUS 网络 PROFIBUS（1），波特率以及其他设置都要与从站相同，否则主站和从站不能实现通信。

在主站 DP 属性设置对话框中切换到“工作模式”选项卡，勾选“DP 主站”复选框，如图 9-39 所示。



图 9-38　从站通信接口区

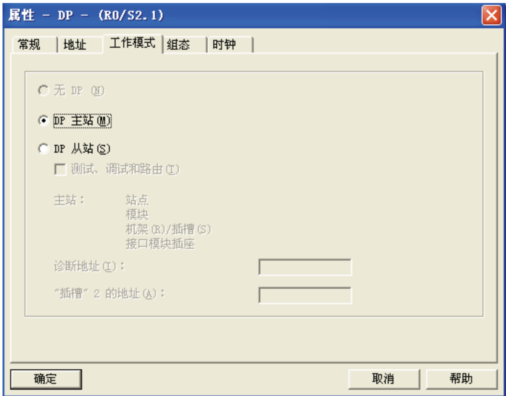


图 9-39　设置主站的 DP 模式

在硬件组态窗口中单击菜单栏中的“视图”→“目录”，在“PROFIBUS DP”下选择“Configured Station”文件夹，将“CPU 31X”拖到主站系统的 DP 接口的 PROFIBUS（1）总线上，这时会同时弹出 DP 从站连接属性对话框，如图 9-40 所示，选择所连接的从站后，单击“连接”按钮确认，这样从站就可以挂接在主站上了，如图 9-41 所示。

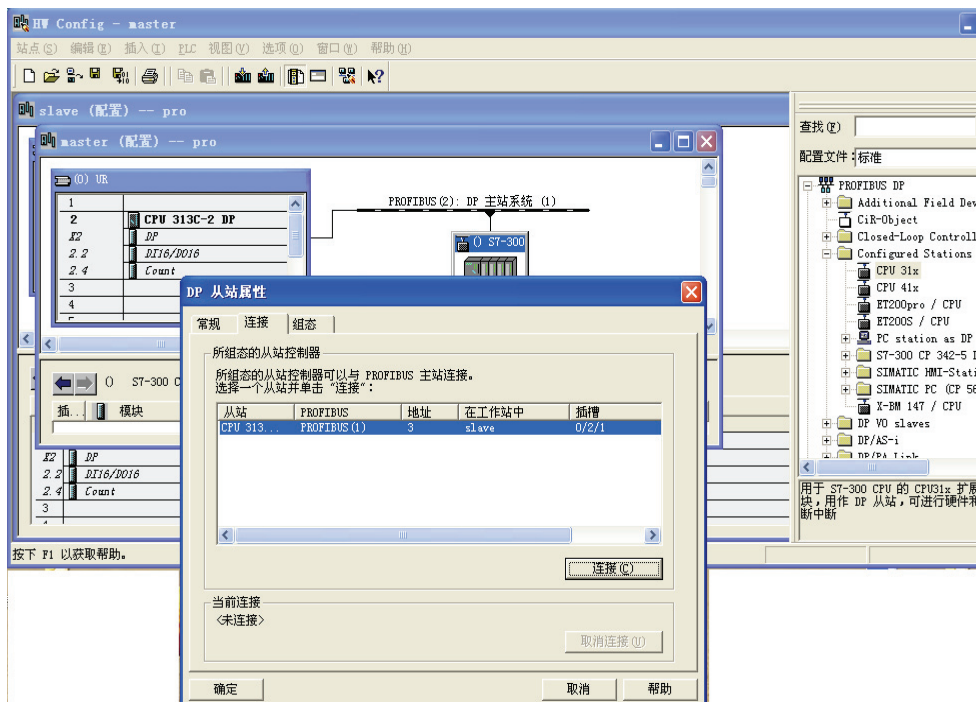


图 9-40　主站与从站的连接

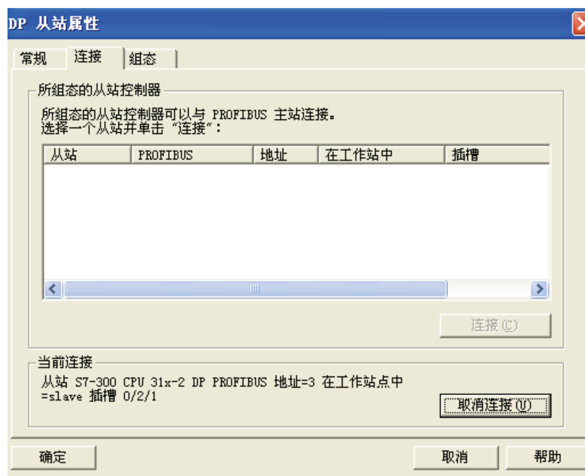


图 9-41　主站与从站连接后的界面

连接完成后，单击“组态”选项卡，设置主站的通信接口区，使得从站的输出区与主站的输入区相对应，从站的输入区与主站的输出区相对应；通信接口设置如图 9-42 所示，

图 9-43 所示为主/从站通信接口区设置完成的界面。

当主/从站硬件组态完成后分别将组态界面下载至对应的 PLC 中。

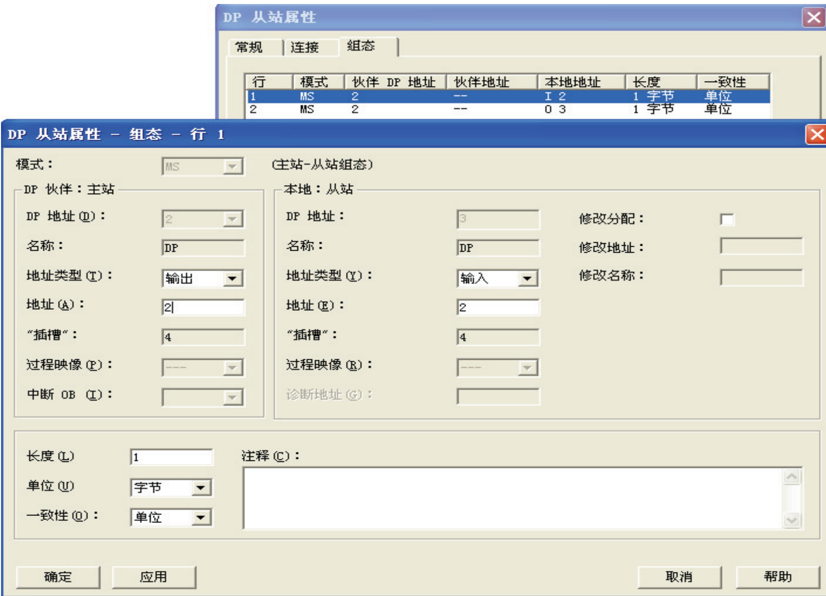


图 9-42　编辑通信接口区

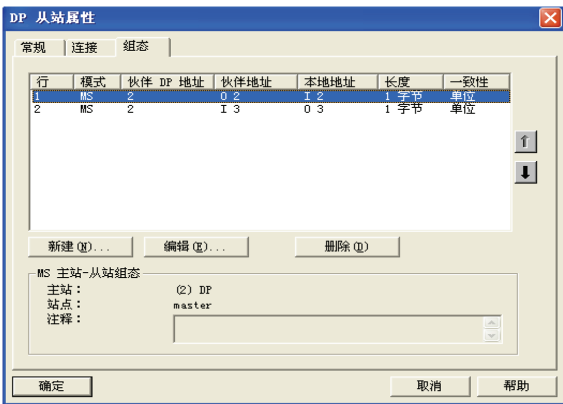


图 9-43　主/从站组态完成的通信数据区

4. 程序设计

根据控制要求，主、从站 PLC I/O 地址分配见表 9-5。

表 9-5a　主站 PLC 的 I/O 地址分配表

PLC 的 I/O 地址	连接的外部设备	在控制系统中的作用
I1.0	按钮 SB1	从站电动机起动按钮
I1.1	按钮 SB2	从站电动机停止按钮
Q1.0	接触器线圈 KM1	主站电动机 M1 工作

表 9-5b　从站 PLC 的 I/O 地址分配表

PLC 的 I/O 地址	连接的外部设备	在控制系统中的作用
I1.0	按钮 SB3	主站电动机起动按钮
I1.1	按钮 SB4	主站电动机停止按钮
Q1.0	接触器线圈 KM2	从站电动机 M1 工作

主站与从站之间信息交换如图 9-44 所示。

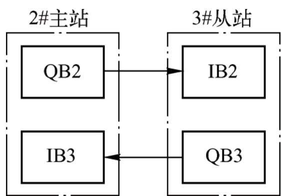


图 9-44　主站与从站之间的数据传送图

主站程序设计如图 9-45 所示，从站程序设计如图 9-46 所示。

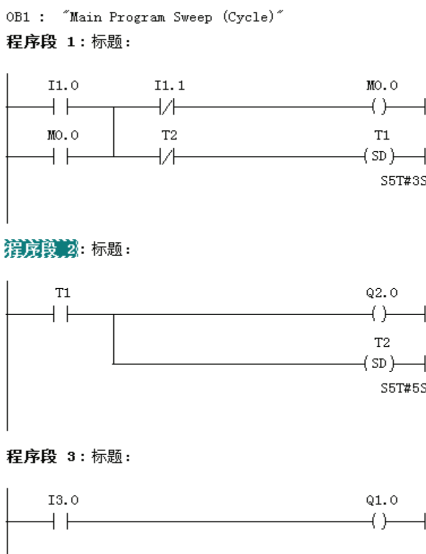


图 9-45　主站程序的实现

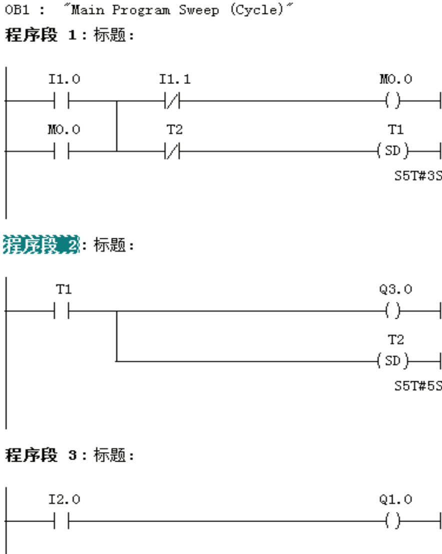


图 9-46　从站程序的实现

9.2.6　S7-300 PLC 与 MM420 变频器的 PROFIBUS-DP 通信

1．MM420 变频器

为了连接进入 PROFIBUS-DP 网络，变频器必须采用 PROFIBUS 模板，这一模板安装在变频器的正面，通过 RS-485 串行接口与变频器进行通信。

PROFIBUS 模板的特点如下：

1）采用 PROFIBUS-DP 协议最多可以控制 125 台变频器（带有重发器）。

2）具有连接 SIMOVIS 或其他维修工具的非周期通信通道。

3）使用 PROFIBUS 控制命令 SYNC 和 FREEZE。

4）使用 S7 管理软件，或其他任何专用的 PROFIBUS 调试工具，系统配置十分方便。

5）不需要单独的供电电源，模板从变频器正面插入，操作十分方便。

6）处理数据的响应时间为 5ms。

7）所有的变频器参数都可以通过串行链路进行访问。

选择带 DP 接口的 MM420 变频器作从站，相关参数如下：

1）变频器货号：6SE6420-2UC11-2AA1。

2）PROFIBUS 模板货号：6SE6400-1PB00-0AA0。

3）输入电压范围：200~240V/（+10%~−10%）。

4）输出电压范围：0~输入电压，三相交流。

5）输出频率范围：0~650Hz。

6）适配电动机功率：0.12kW。

2．PROFIBUS 总线系统硬件的连接和参数配置

1）正确连接主站和变频器之间的总线电缆，包括必要的终端电阻和各段网络。

2）总线电缆必须是屏蔽电缆，其屏蔽层必须与电缆插头/插座的外壳相连。

3）变频器的从站地址（参数 P0918）必须正确设置，使其与 PROFIBUS 主站配置的从站地址相一致；也可以在变频器的通信板（CB）上完成，通信板（CB）上有一排拨钮用于设置地址，如图 9−47 所示，每个拨钮对应于“8−4−2−1”码的数据，所有的拨钮处于“ON”位置对应的数据相加的和就是站地址； 例如如果设置变频器站地址为 4， 则将 3 位置的拨钮置在“ON”的状态；拨钮开关为“0”时，P0918 指定的地址有效，不为“0”时，拨钮的设定值优先。变频器站地址必须和 STEP7 软件中硬件组态的地址保持一致，否则不能通信。

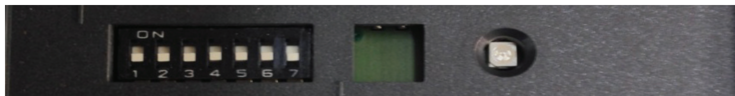


图 9−47 通信板上的拨钮开关

4）变频器常用操作模式见 5.4.2 节

本例中变频器设置步骤为先设定参数 P0010 = 30，P970 = 1，把其他参数复位；然后设定参数 P0010 = 0，P0700 = 6，P1000 = 6，使变频器处于 PROFIBUS 总线控制状态。

3．硬件组态

（1）PROFIBUS-DP 网络的建立

在硬件组态中，双击“DP”，弹出“属性−DP”界面，如图 9−48 所示，可见接口状态。

单击图中“属性”按钮，弹出图 9−49 界面，选择 PLC PROFIBUS-DP 地址：2，单击“新建”按钮，设置新建网络参数，如图 9−50 所示，单击“确定”按钮，弹出 9−51 界面。

单击图 9−51“确定”按钮，弹出图 9−52 所示界面，单击“确定”按钮，弹出图 9−53 所示界面，该 PLC 是作为 PROFIBUS-DP 主站设置的，从图 9−54 可见其性能。

（2）MM420 从站的组态

选择目录“PROFIBUS DP”→“SIMOVERT”→“MICROMASTER 4”，如图 9−55 所示。

选择 MM420 的地址为“4”和工作方式为“0PKW，2PZD（PPO 3)”，得到 MM420 从

站配置，通信报文格式的含义是报文中有 0 个字的 PKW（参数标识值），有 2 个字的 PZD（过程数据）。

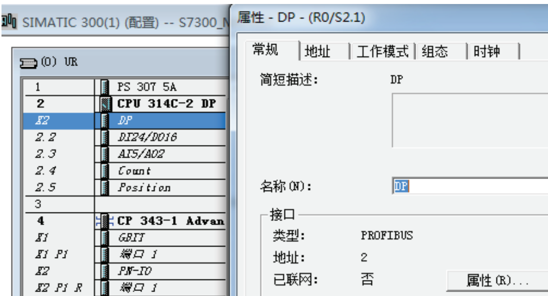


图 9-48　DP 属性界面

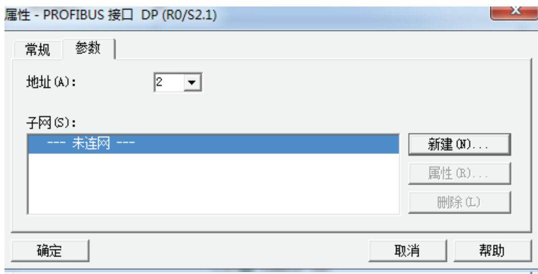


图 9-49　PROFIBUS 接口设置

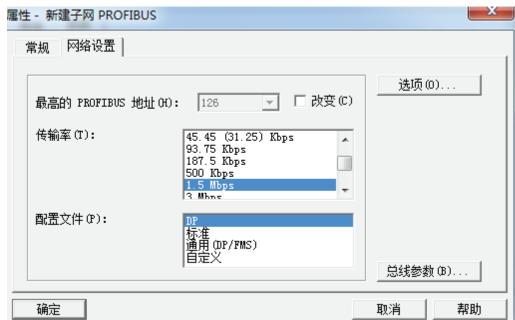


图 9-50　网络参数选择

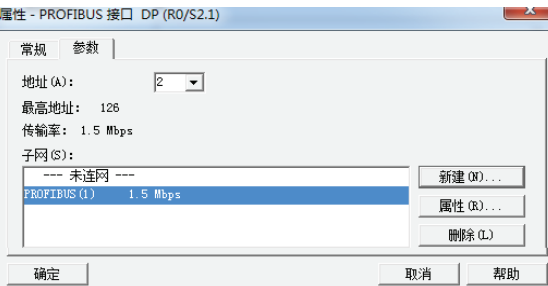


图 9-51　建立 PROFIBUS（1）网络

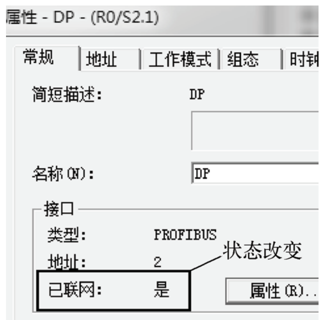


图 9-52　接口属性

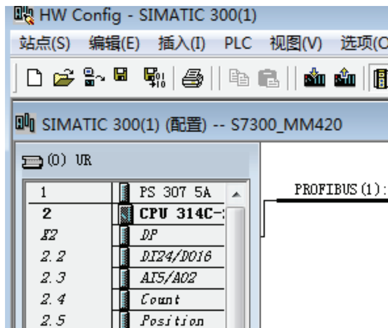


图 9-53　建成 PROFIBUS（1）网络



图 9-54　DP 站工作模式的选择

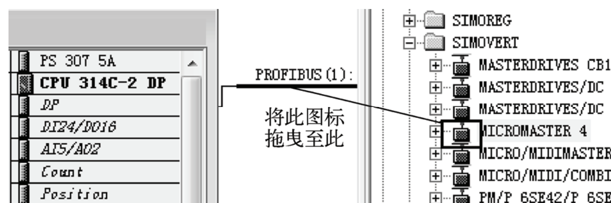


图 9-55 MM420 硬件组态

由 PKW 和 PZD 组成多种 PPO 形式，可根据设计需要选择合适的形式。PKW 用于读/写参数值，用于读/写变频器中的某个参数；PZD 是为控制和监测变频器而设计的，如果要控制变频器起停、设定频率等参数则需要用到 PZD，过程数据一直被传输，具有最高的优先级和最短的间隙，其数据根据传送方向的不同而不同：当数据由主站传向变频器时，PZD 区由控制字 STW 和频率设定值 HSW 构成；当数据由变频器传向主站时，PZD 区由返回变频器的状态字 ZSW 和实际速度值 HIW 构成。

控制字与状态字含义见表 9-6。

表 9-6a 控制字含义说明（STW）

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
故障确认	设定值使能	RFG 开始	RFG 使能	脉冲使能	OFF3	OFF2	ON/OFF1
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bi8
—	电位计降速	电位计升速	—	设定值反向	PLC 控制	点动向左	点动向右

例如：QW=16#047E，表示停止；QW=16#047F，表示起动/前进；QW=16#0C7F，表示起动/后退.。

表 9-6b 状态字含义说明（ZSW）

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
驱动警告激活	激活禁止合闸状态	OFF3 激活	OFF2 激活	驱动故障	驱动正在运行	驱动就绪等待运行	驱动就绪
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bi8
变频器过载	电动机顺时针运行	电动机过载	电动机保持制动激活	电动机最大电流警告	达到最大频率	PZD 控制	设定值/实际值偏差

输入/输出模块的地址已由 STEP 7 软件自动分配，如图 9-56 所示。主站读取 MM420 数据的存放地址设置为 IB290~IB293，共 2 个字；主站向 MM420 写数据的存放地址设置为 Q272~Q275，共 2 个字，最后编译并保存组态完成的硬件。

4．控制功能的实现

在变频器与 PLC 数据传输过程中最常用的是通过调用系统功能块实现变频器与 PLC 的通信，在 PLC 的 OB1 中调用系统功能 SFC14/SFC15，完成对 MM420 数据的读/写功能。其中系统功能块 SFC14 的作用是给数据解包，将变频器数据读到 PLC，系统功能块 SFC15 的作用是给数据打包，将 PLC 数据写入变频器。

项目结构如图 9-57 所示。

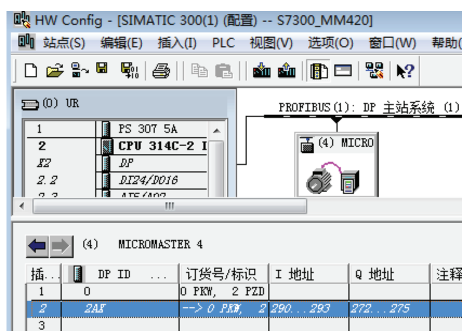


图 9-56　变频器参数设置

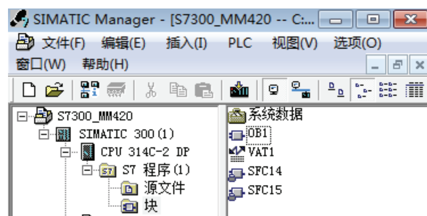
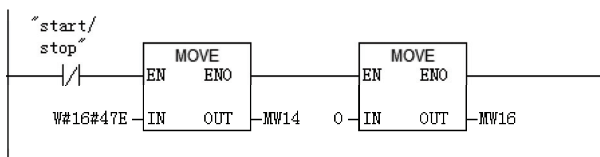


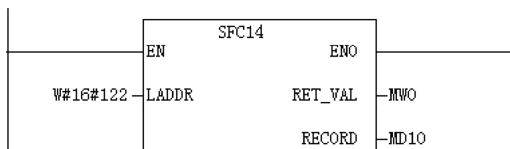
图 9-57　项目结构

OB1 程序设计如图 9-58 所示。

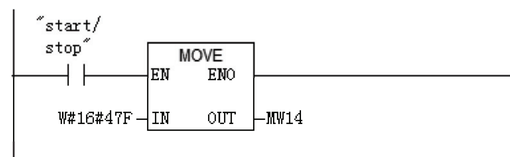
程序段 1：MW14为控制字，MW16为设定频率



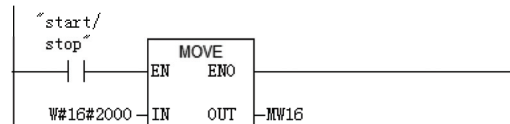
程序段 2：SFC14读取MM420的应答数据，硬件地址为290（换算成十六进制为122）



程序段 3：起动变频器



程序段 4：起动后的频率为25Hz



程序段 5：SFC15为主站向MM420写数据，硬件起始地址为272（换算成十六进制为110）

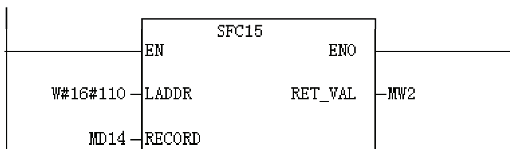


图 9-58　OB1 程序

MD10 为主站从变频器读到状态字和实际速度值，MD14 为主站向从站写入的控制字及频率给定值，图 9-59 为这两个变量在停止、起动及稳定运行时的变化情况。

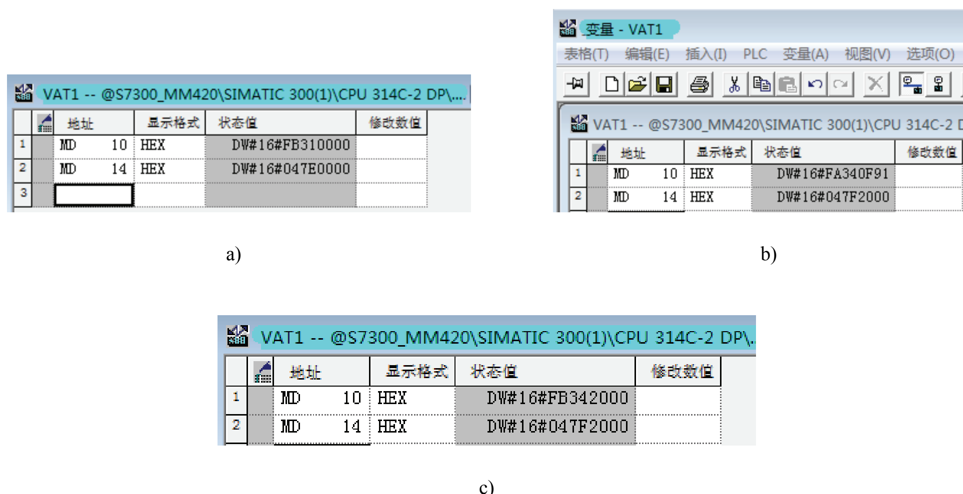


图 9-59　MD10 和 MD14 变量在变频器不同运行状态下的变化情况

a) 变频器停止状态　b) 变频器起动过程中　c) 变频器运行中

在设定变频器频率时需要注意实际频率与通信变量之间的对应关系，见表 9-7。

表 9-7　变频器实际频率与通信变量之间的对应关系

频率实际值/Hz	十六进制对应值	十进制对应值
50	4000	16384
37.5	3000	12288
25	2000	8192
18.75	1500	5376
12.5	1000	4096
6.25	500	1280

9.3　工业以太网协议及其应用

9.3.1　工业以太网基础知识

由于工业控制市场存在多种现场总线标准并存的现象，且短期内无法统一；从用户应用的角度来看，多种现场总线标准并存会导致在一个具体应用系统中涉及多种不同标准的现场总线设备，需要解决不同标准系统之间的相互连接和相互操作的问题。为了解决该问题，出现了工业以太网技术。

工业以太网是工业环境中一种有效的子网，既适用于管理级又适用于单元级，传输速率为 10~100Mbit/s。在自动化领域，越来越多的企业需要建立包含从工厂现场设备层到控制层、管理

层等各个层次的综合自动化网络管控平台，建立以工业控制网络技术为基础的企业信息化系统。

工业以太网是以太网技术向控制网络延伸的产物，是工业应用环境下信息网络与控制网络的结合。目前对工业以太网没有严格的定义，各家推出的工业以太网在技术上也存在相当大的差距，一般来讲，工业以太网是指技术上与商用以太网（IEEE802.3 标准）兼容，在产品设计时，材质的选用、产品的强度以及网络的实时性、可靠性、抗干扰性、安全性等方面均能满足工业现场要求。

1．以太网

以太网进入工业控制领域无论是价格还是技术都具有一定的优势。

1）价格优势：由于信息网络的存在和以太网的大量使用，使其具有价格明显低于控制网络相应软硬件的特点，如通过普通网卡就可将计算机连接到工业以太网络中。

2）技术优势：技术成熟、易于得到、已为许多人掌握，有利于企业网络的信息集成、便于上层网络的连接。

以太网最早来源于 Xerox 公司于 1973 年建造的网络系统，是一种总线型局域网，以基带同轴电缆作为传输介质，采用 CSMA/CD 协议。其核心思想是使用共享的公共传输信道，即遵循 IEEE802.3 标准，可以在光缆和双绞线上传输的网络。以太网也是当前主要应用的一种局域网（Local Area Network，LAN）类型。目前的以太网按照传输速率大致分为以下四种：

1）10Base-T 以太网：传输速率为 10Mbit/s，传输介质是双绞线。

2）100Base 以太网：也称快速以太网，传输速率为 100Mbit/s，采用光缆或双绞线作为传输介质，兼容 10Base-T 以太网。

3）Gigabit 以太网：扩展的以太网协议，传输速率为 1Gbit/s，采用光缆或双绞线作为传输介质，基于当前的以太网标准，兼容 10Mbit/s 以太网和 100Mbit/s 以太网。

4）10 Gigabit 以太网：是一种速度更快的以太网技术，传输速率达到百亿比特每秒，采用光缆作为传输介质。主要用于局域网（LAN）、广域网（WAN）以及城域网（MAN）之间的相互连接。

随着信息技术的发展，信息交换技术涉及领域越来越广，控制网络与普通计算机网络、Internet 的联系也越来越密切。控制网络技术需要考虑与计算机网络连接的一致性，需要提高对现场设备通信能力的要求，这些都是控制网络设备的开发者与制造商把目光转向以太网技术的重要原因。

工业网络与传统的办公室网络相比，有其自己的要求和特点，见表 9-8。

表 9-8　工业网络与传统办公室网络的比较

比 较 项 目	办公室网络	工 业 网 络
应用场合	普通办公场合	用于工业现场，环境恶劣，存在各种干扰
用途	办公、通信	设备通信、远程监控、故障诊断等
拓扑结构	支持线形、环形、星形等结构	支持线形、环形、星形等结构，便于各种结构的组合和转换，具有一定的柔性安装和扩展能力
实时性	一般的实时性需求	对数据传输的快速性和系统响应的实时性要求高
信息特征	信息量大且综合、多样	数据量小
网络监控和维护	网络监控必须有专职人员使用专用工具完成	网络监控成为工厂监控的一部分，网络模块可以被 HMI 软件监控，故障模块易于更换

可见，以太网要用于工业控制中，在设计与制造时必须充分考虑并满足工业网络应用的需求。以太网技术经过多年的发展，特别是在Internet中的广泛应用，使得其更为成熟，并得到了广大开发商与用户的认同。因此无论从技术上还是产品价格上，以太网较之其他类型网络技术都具有明显的优势。

但作为传统上用于办公室和商业用途的以太网，在引入工业控制领域时，仍然由于以太网的不确定性引起了很大的争议，存在着怎样解决工业控制网络的实时性和可靠性问题。

2．以太网应用于工业现场的关键问题

（1）通信的实时性

以太网采用 CSMA/CD 的总线访问机制，遇到碰撞时无法保证信息及时发送出去，这种平等竞争的介质访问控制方式不能满足工业自动化领域对通信的实时性要求，因此需要有针对这一问题的切实可行的解决方案。

（2）对环境的适应性与可靠性

以太网是按办公环境设计的，将它用于工业控制环境，其环境适应能力、抗干扰能力等是许多从事自动化的专业人士所关注的问题。例如像RJ45一类的连接器，在工业上应用非常容易损坏，应该采用带锁紧机构的连接件，使设备具有更好的抗振动、抗疲劳能力；在产品的设计时要考虑各种环境因素，使得参数能满足工业现场的要求。

（3）总线供电

在控制网络中，现场控制设备的位置分散性使得它们对总线有提供工作电源的要求。现有的许多控制网络技术都可以利用网线对现场设备供电。工业以太网目前没有对网络节点供电作出规定。一种可能的方案是利用现有的 5 类双绞线中另一对空闲线供电。一般在工业应用环境下，要求采用直流 10~36V 低压供电。

（4）本质安全

工业以太网如果要用在一些易燃易爆的危险工业场所，就必须考虑本安防爆问题，这是在总线供电解决之后要进一步解决的问题。

以太网技术用于工业现场虽然存在上述一些问题，但并不意味着以太网就不能用于现场控制层，事实上以太网在很多对时间要求不是非常苛刻的现场层，已有很多成功应用范例。而且随着以太网技术的发展和标准的进步，以太网在工业环境中应用存在的问题正在逐渐完善和解决，例如，采用专用的工业以太网交换机、定义不同的以太网帧优先等级，使用户所希望的信息能够以最快的速度传递出去；网络采用双绞线电缆、光缆等传输介质，以提高网络的抗干扰能力和可靠性。

事实上，在工业数据通信与控制网络中，直接采用以太网作为控制网络的通信技术只是工业以太网发展的一个方面，现有的许多现场总线控制网络都提出了与以太网结合用以太网作为现场总线网络的高速网段，从而使控制网络能与 Internet 融为一体。

在控制网络中采用以太网技术无疑有助于控制网络与互联网的融合，使控制网络无须经过网关转换即可直接与互联网连接，使测控节点也能成为互联网络的节点。在控制器、测量变送器、执行器、I/O 卡等设备中嵌入以太网通信接口，嵌入 TCP/IP 协议，嵌入 Web Server 便可形成支持以太网、TCP/IP 协议和 Web 服务器的 Internet 现场节点。在应用层协议尚未统一的环境下，借助 IE 等通用的网络浏览器实现对生产现场的监视与控制，进而实现远程监控，也是人们提出且正在实现的一个有效的解决方案。

3．实时以太网

根据设备应用场合，按照实时性要求将工业自动化系统划分为如下 3 个范围：

1）信息集成和较低要求的过程自动化应用场合，实时响应时间要求为 100ms 或更长。

2）绝大多数的工厂自动化应用场合实时响应时间的要求最少为 5~10ms。

3）对于高性能的同步运动控制，特别是在 100 个节点以下的伺服运动控制应用场合，实时响应时间要求低于 1ms，同步传送和抖动小于 1μs。

研究表明，工业以太网的响应时间可以满足绝大多数工业过程的控制要求，但对于响应时间小于 4ms 的应用，工业以太网已不能胜任。为了满足工业控制实时性能的要求，各大公司和标准组织纷纷提出各种提升工业以太网实时性的技术解决方案，这些方案建立在 IEEE802.3 标准的基础上，通过对其相关标准的实时扩展，来提高实时性，并且做到与标准以太网的无缝连接，这就是实时以太网（Real Time Ethernet，RTE）。

为了规范 RTE 工作的行为，2003 年 5 月，IEC/SC65C 专门成立了 WG11 实时以太网工作组，负责制定 IEC61784-2“基于 ISO/IEC8802.3 的实时应用系统中工业通信网络行规”国际标准，在该标准中包括 Ethernet/IP、PROFINET、P-NET、Interbus、VNET/IP、TCNET、EtherCAT、Ethernet Powerlink、EPA、Modbus/TCP 及 SERCOS 等 11 种实时以太网行规集。

例如我国制定的“用于工业测量与控制系统的 EPA 系统结构与通信标准”，规定了网络的时间同步精度为 8 个等级，等级如下：

0—无精度要求；1—时间同步精度<1s；2—时间同步精度<100ms；3—时间同步精度<10ms；4—时间同步精度<1ms；5—时间同步精度<100μs；6—时间同步精度<10μs；7—时间同步精度<1μs。

据美国权威调查机构 ARC（Automation Reserch Company）报告，今后 Ethernet 不仅继续垄断商业计算机网络和工业控制系统的上层网络通信市场，也必将领导未来现场控制设备的发展，Ethernet 和 TCP/IP 将成为器件总线和现场设备总线的基础协议。

9.3.2　工业以太网的现状与发展前景

1．工业以太网的现状

（1）技术改造

针对以太网在工业现场使用中面临的问题，现已出现了不少解决方案。如对以太网的数据链路层进行改进来实现实时性和确定性的要求，当然对其只是小改进或者变相的改进，而不能改动最基本的 CSMA/CD 介质访问方式，否则就改变了以太网的性质；采用能满足工业现场要求的连接方式来满足现场设备的安装和可靠性要求；至于在以太网上实现总线供电和防爆等技术还在进一步的开发和研究中。

（2）大公司增加开发力度

许多大公司都提出了工业以太网的实现方案，并且也陆续推出了自己的产品。比较有影响的如 FF 的高速以太网 HSE、ROCKWELL 公司的以太网工业协议 Ethernet/IP、SIEMENS 公司的 PROFINET、SCHNEIDER 公司的 Modbus/TCP，以及 IDA 集团的分布式自动化接口（Interface for Distributed Automation，IDA）等。

（3）几种主要的工业以太网

1）Ethernet/IP：由 ODVA、CI 和 IEA 三个国际组织于 2000 年联合推出，ROCKWELL

公司是其主要支持者。它基于以太网技术、TCP/IP 技术和 CIP（Control and Information Protocol）技术，因此兼具工业以太网和 CIP 网络的优点。

2）高速以太网（HSE）：现场总线基金会（FF）于 2000 年发布了 HSE 的技术规范，定位于实现控制网络与 Internet 的集成，由 HSE 链接设备将 H1 网段信息传送到以太网的主干上并进一步送到企业的 ERP 和管理系统。

3）PROFINET：由德国西门子公司于 2001 年发布。PROFINET 的基础是组件技术，每一个设备都被看成是具有 COM 接口的自动化设备，简化了编程。PROFINET 是中国国家标准，标准号：GB/Z 20541-2006。

4）Modbus/TCP：由 SCHNEIDER 公司于 1999 年公布，以一种非常简单的方式将 Modbus 框架潜入到 TCP/IP 结构中，基本上没有对 Modbus 协议本身进行修改，只是为了满足控制网络实时性的需要，改变了数据的传输方法和通信速率。Modbus/TCP 是中国国家标准，标准号：GB/Z 19582.3-2008。

5）EtherCat：由德国倍福 BECKHOFF 公司开发，并由 EtherCAT 技术组（EtherCAT TechnologyGroup，ETG）支持。它采用以太网帧，并以特定的环状拓扑发送数据；EtherCAT 保留了标准以太网功能，并与传统 IP 协议兼容。

2．工业以太网的发展前景

由于各个国家各个公司的利益之争，虽然早在 1984 年国际电工技术委员会/国际标准协会（IEC/ISA）就开始着手制定现场总线的标准，但至今仍未完成统一的标准；很多公司也推出其各自的现场总线技术，但彼此的开放性和互操作性还难以统一。随着以太网技术的不断发展与进步，其有着逐步取代传统现场总线的趋势；以太网的传输速率以及其标准化和开放性对于有高速传输要求的现场总线也是一种提速，而工业自动化产品制造商也同样对以太网技术融合在不断进行探索。

因此可以说，工业以太网在工业通信网络中的使用将构建从底层的现场设备到控制层、企业管理决策层的综合自动化网络平台，从而消除企业内部的各种自动化孤岛。以太网作为 21 世纪未来工业网络的首选，将在控制层和现场设备级成为标准的高速工业网络，有着广泛的应用和发展前景。

以太网所具有的低成本、全开放、传输速率高及应用广泛等优点，使它在工业控制系统应用中拥有无可比拟的优势。应用于工业控制现场的固有缺陷如通信实时性、可互操作性、网络安全性等都已得到了很大程度的改善。

工业以太网用于控制网络的优势：

1）易于实现设备通信及远程访问：基于 TCP/IP 的以太网采用国际主流标准，协议开放，容易实现不同厂商设备之间的互连和互操作，容易实现远程访问和远程诊断等功能。

2）系统配置灵活：不同的传输介质可以灵活组合，支持冗余连接配置，不会因系统增大而出现不可预料的故障，有成熟可靠的系统安全体系。

但由于以太网还不能够完全解决实时性和确定性问题，大部分现场层仍然会首选现场总线技术。由于技术的局限和各个厂家的利益之争，这样一个多种工业总线技术并存、以太网技术不断渗透的现状还会维持一段时间，用户可以根据技术要求和实际情况来选择所需的解决方案。

9.3.3　S7-300 PLC 与 ET200S 的 PROFINET 通信

1．PROFINET 技术介绍

PROFINET 是 PI（PROFIBUS International，PI）推出的开放性标准，用于实现基于工业以太网的集成自动化方案，其标准涵盖了控制器各个层次的通信，其中包括 I/O 设备的普通自动控制领域和功能更加强大的运动控制领域。

德国西门子公司 1998 年发布的工业 Ethernet 白皮书与 2001 年发布的工业 Ethernet 的规范，统称为 PROFINET。PROFINET 基于工业以太网技术，使用 TCP/IP 和 IT 标准，是一种实时以太网技术，同时它无缝地集成现有的现场总线系统，从而实现现有的现场总线技术与工业以太网的有机融合。

作为国际标准IEC61158的重要组成部分，PROFINET是完全开放的协议，而且PROFINET和标准以太网完全兼容，集成 IRT（即等时实时通信）功能的交换机和一个普通交换机在平时工作时是完全一样的；也就是说，IRT 交换机可以和普通交换机一样使用，即使在使用实时通道时，同样可以在开放通道使用其他标准功能。

为了给不同类型的自动化应用提供最佳的技术支持，PROFINET 标准提供了两种基于工业以太网的自动化集成解决方案：PROFINET I/O（分布式 I/O）、PROFINET CBA（基于组件的分布式自动化系统）。

其中，PROFINET I/O 是使用以太网连接和 PROFINET 通信的分散的外围设备，PROFINET I/O 关注的是采用简单的通信设备实现适合的数据传输；PROFINET CBA 以工艺技术模块的面向对象的模块化为基础，这些模块的功能采用统一的 PROFINET 定义方式进行封装，它满足成套构造者和操作者对于系统级的工程设计过程而与制造商无关的要求。使用 PROFINET，能使简单的分布式 I/O 和严格时间要求的应用以及基于组件的分布式自动化系统集成到以太网通信中。

2．硬件配置及介绍

CPU315 2PN/DP PLC 1 台、PS307 5A 电源模块 1 只、ET200S 模块（配置包括：IM151-3 PN 接口模块、PM-E 24V 电源模板、4DI/DC24V、4DO/DC24V/0.5A）1 块、PROFINET 电缆（包括 RJ45 插头）2 根。

ET200S 为分布式 I/O 系统，其中 IM151-3 PN 为接口模块，可以连接 ET200S 与 PROFINET I/O，带有两个端口的集成交换机，其主要属性如下：

- 可以为装配的电子模块和电动机起动器准备数据。
- 可以为背板总线供电。
- 传送并备份 SIMATIC 微型存储卡上的设备名称。
- 支持的以太网服务包括 ping、arp、网络诊断（SNMP）/MIB-2。
- 中断：包括诊断中断、过程中断、插入/卸下模块中断、维护中断。
- 最大地址空间为 256 个字节的 I/O 数据。
- 最多可操作 63 个模块。

为了更好地使用 IM151-3 PN 接口模块请查看相关手册。

如图 9-60 所示，PROFINET 电缆导线由 4 根绞合线组成，并采用双屏蔽，特别适用于易受电磁干扰的工业环境中。RJ45 插头具有坚固的金属外壳和集成绝缘穿刺触点，可用于连接

工业以太网电缆；用于连接非交叉 100Mbit/s 以太网，距离最远 100m，无须使用插接线；如图 9-61 所示，打开金属外壳，里面有彩色标记，能方便地将芯线与端子连接，PROFINET 电缆线很容易与 IE FC 接头的绝缘刺破触点连接，无须专门工具。



图 9-60　PROFINET 电缆

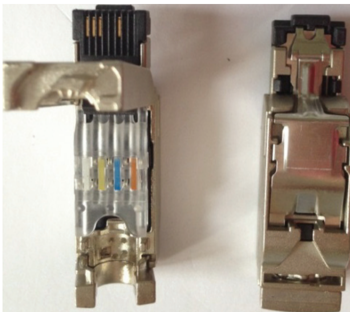


图 9-61　RJ45 插头

3．系统硬件组态

打开 SIMATIC Manager 编程软件的 Project manage 界面，新建“test”项目，如图 9-62 所示，鼠标右键单击“test”→“插入新对象”→“SIMATIC 300 站点”，出现图 9-63 所示界面。

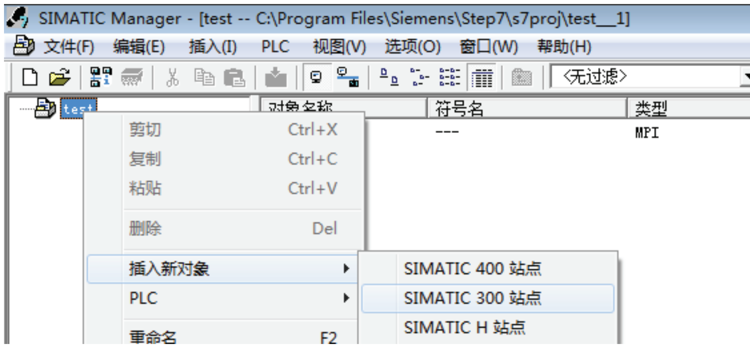


图 9-62　插入 PLC 站点操作步骤

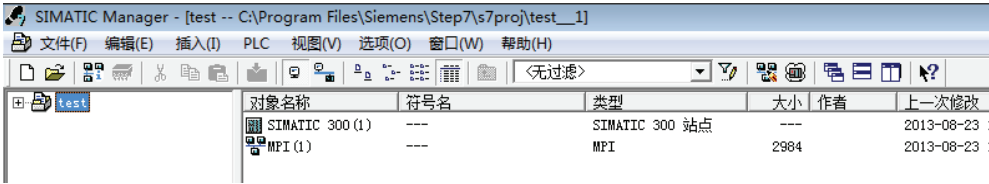


图 9-63　插入 PLC 站点

展开“test”前面的加号，出现图 9-64 所示界面，双击右侧“硬件”，进入硬件组态界面，分别进行机架、电源、CPU 模块及 PROFINET I/O 模块的配置。CPU 模块的配置界面如图 9-65 所示，ET200S 模块的配置过程如图 9-66 所示。

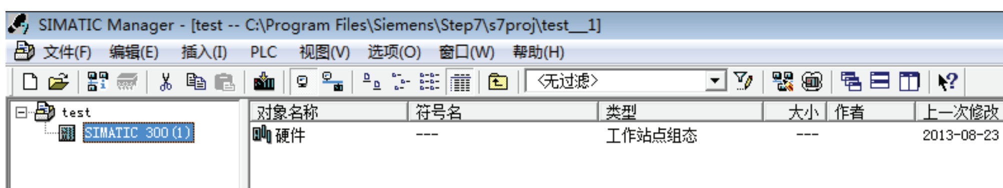
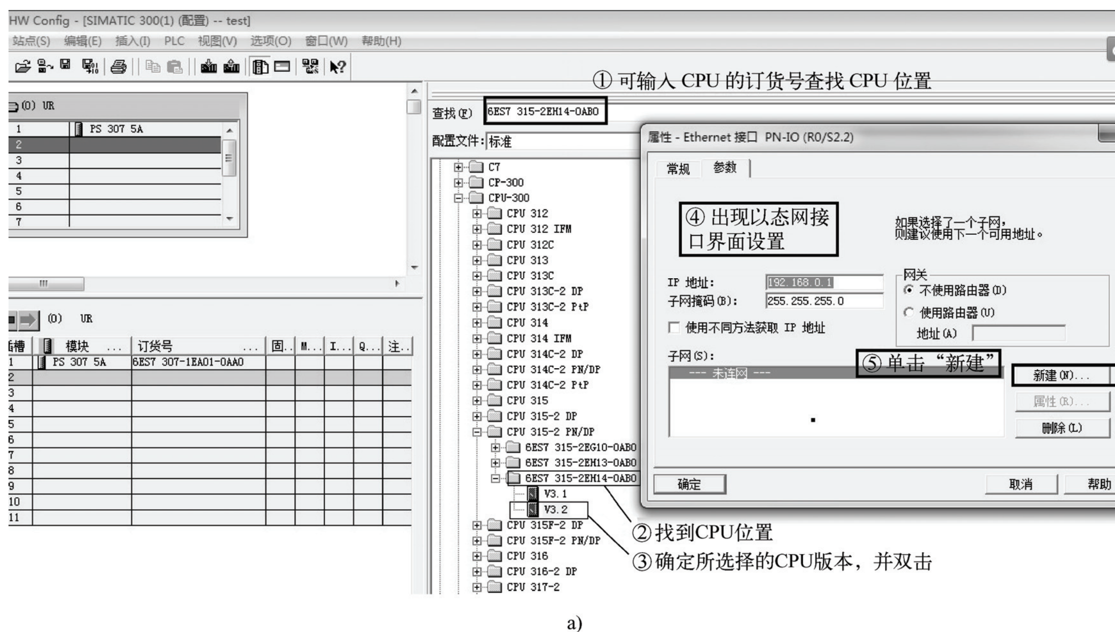
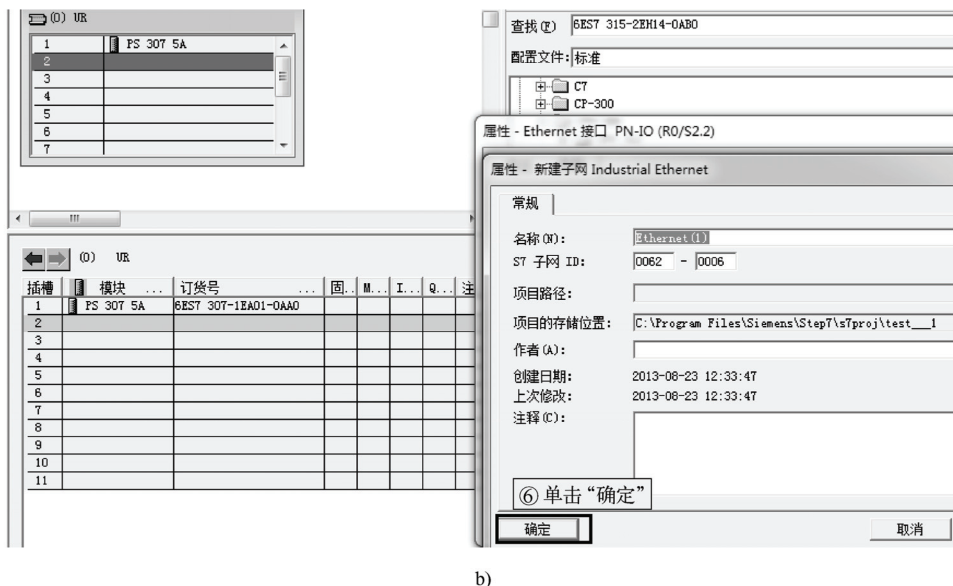


图 9-64　展开左侧 test 菜单栏

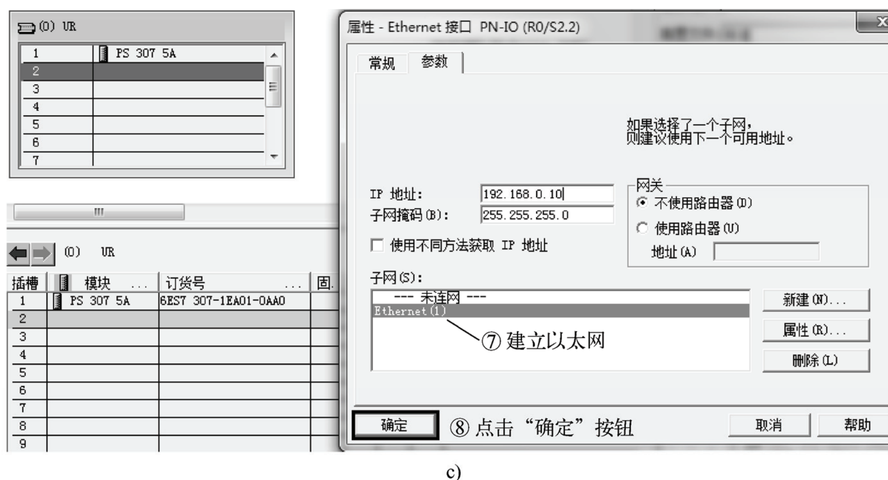


a)

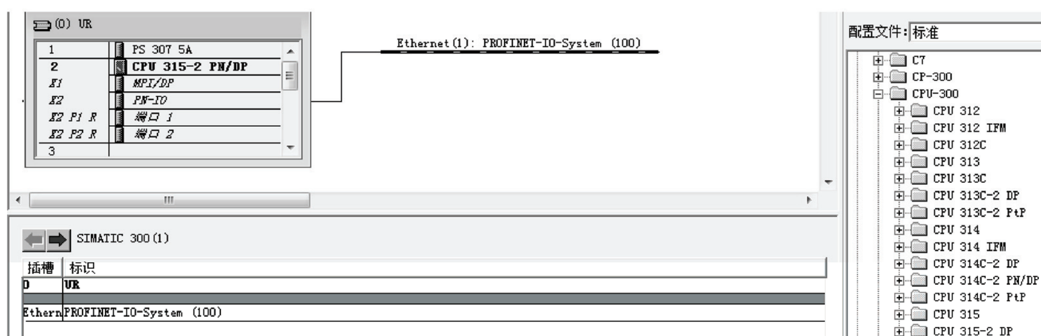


b)

图 9-65　插入 CPU 模块

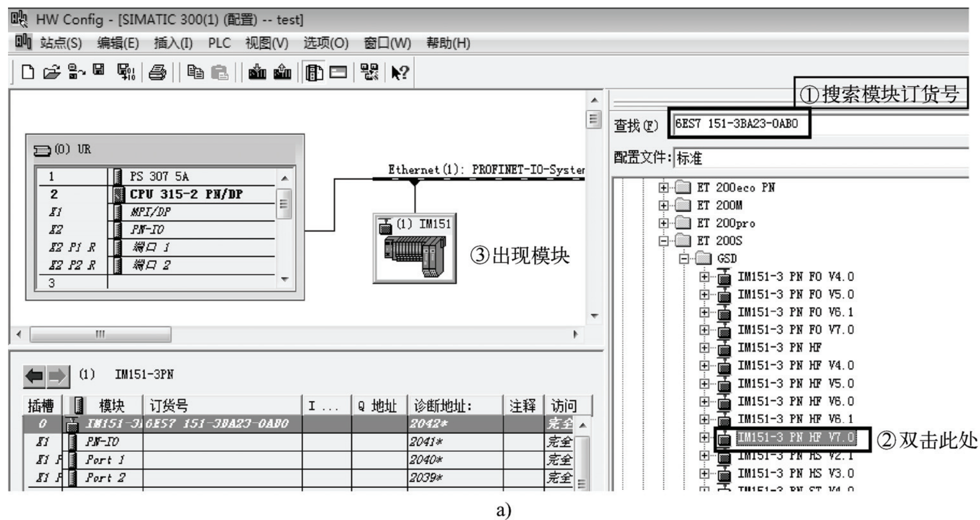


c)



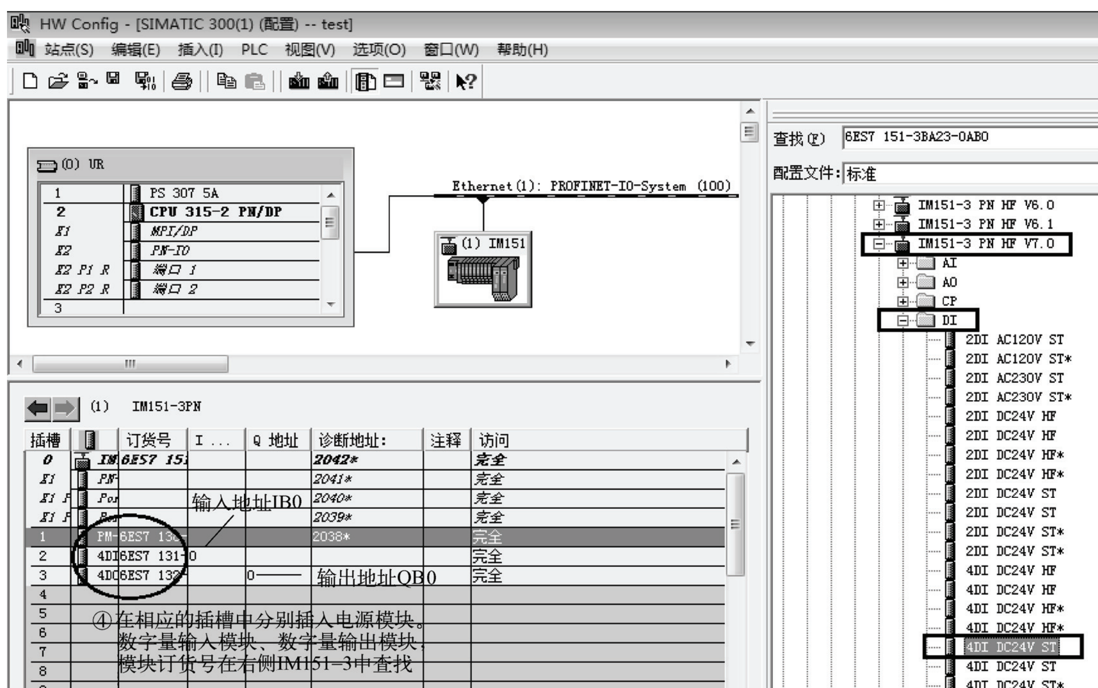
d)

图 9-65　插入 CPU 模块（续）

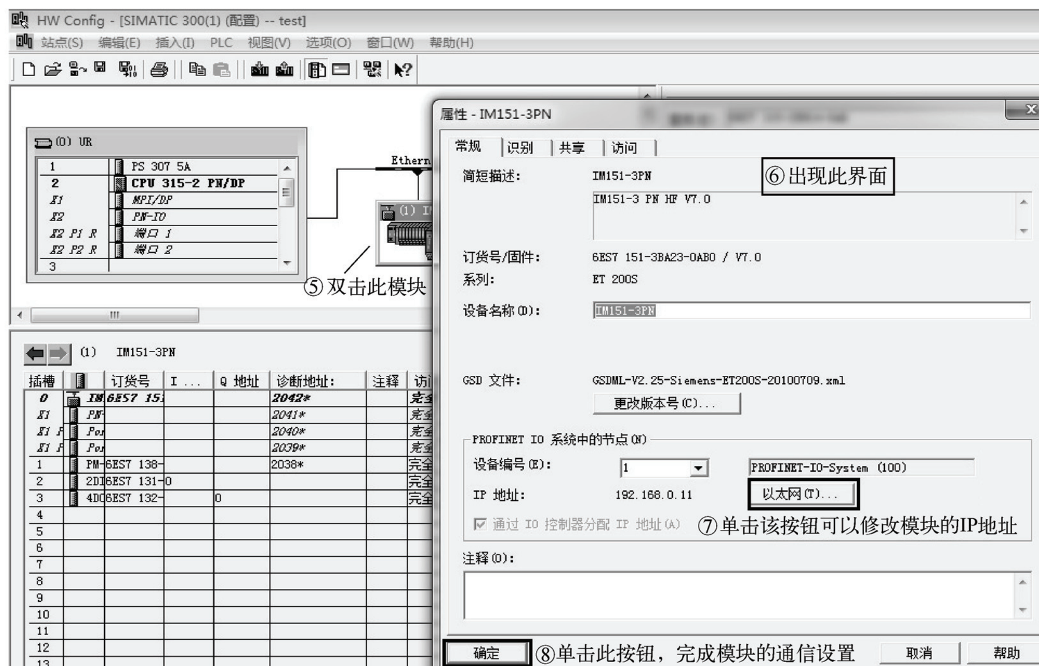


a)

图 9-66　组态 ET200S 模块



b)



c)

图 9-66　组态 ET200S 模块（续）

硬件配置完成后，对组态的硬件配置进行“保存并编译”，系统的硬件组态完成。

4．程序的编写

要求编写的程序实现如下功能：

1）PLC 能读到远程 ET200S 的 4DI 输入模块的输入状态，并将其存储在 MB0 中。

2）远程 ET200S 的 4DO 输出模块的输出状态能随着 PLC 中 MB1 的状态变化而变化。

程序建立步骤如图 9-67 所示，双击“OB1”，进入 LAD/STL/FBD 的编程界面。

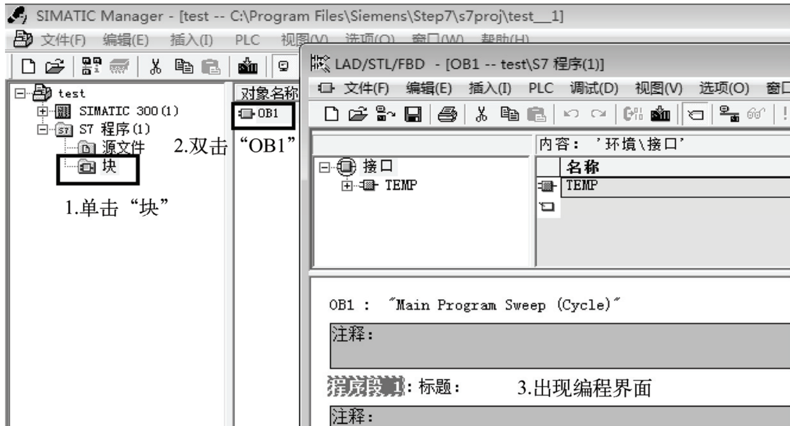


图 9-67　进入 OB1 进行程序编写

梯形图设计如图 9-68 所示，编译并保存。

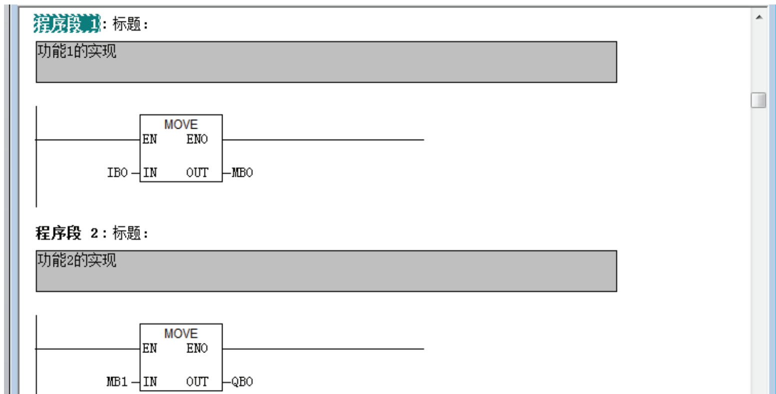


图 9-68　程序的实现

5．设置 PG/PC 的通信接口

本例中，PC 与 PLC 的连接选择 TCP/IP 通信协议。在 SIMATIC Manager 界面中，单击“选项”→“设置 PG/PC 接口”→选择通信协议“TCP/IP→Realtek…”，如图 9-69 所示。

单击“属性”按钮，弹出如图 9-70 所示界面，单击“确认”按钮，完成 PLC 与 PC 的通信设置。

设置完成后，将 PROFINET 电缆一端接入 CPU315 PLC 的一个 PN 接口，另一端插入计算机的 RJ45 接口，即可实现计算机对 PLC 的硬件组态和程序下载、上传和在线监视等功能。

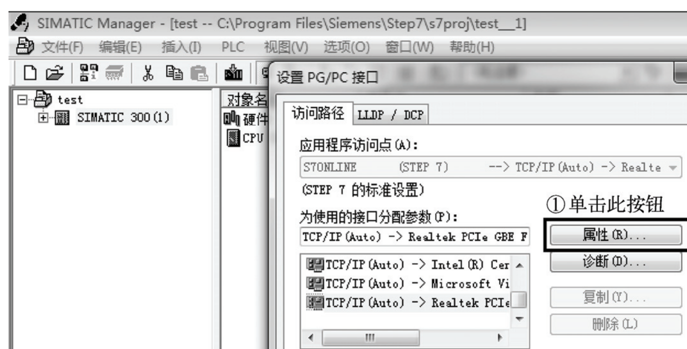


图 9-69　设置 PG/PC 接口的操作界面

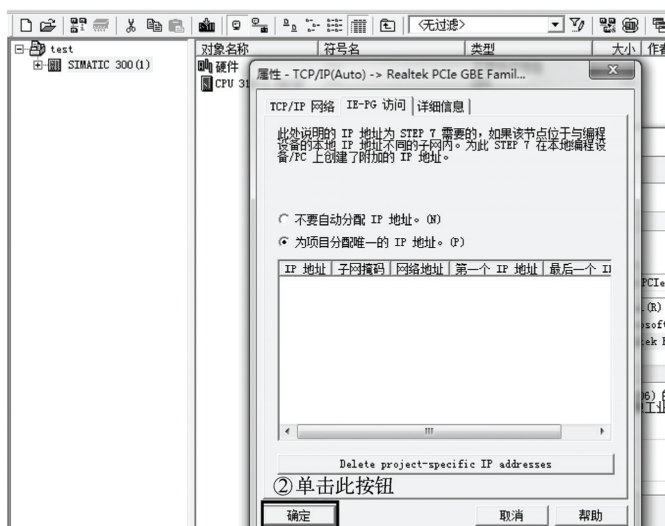


图 9-70　接口属性设置界面

6．将系统硬件组态、用户程序下载至 PLC 中

如图 9-71 所示进行操作，当完成步骤 4 时，下载工作完成。也可以分别在硬件组态界面下载硬件配置和在 OB1 中下载程序。

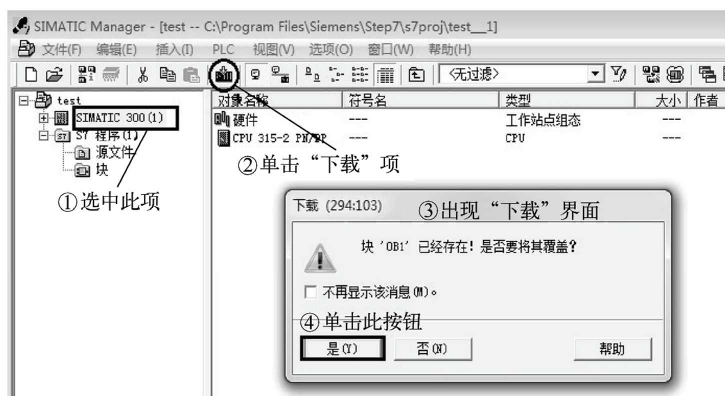
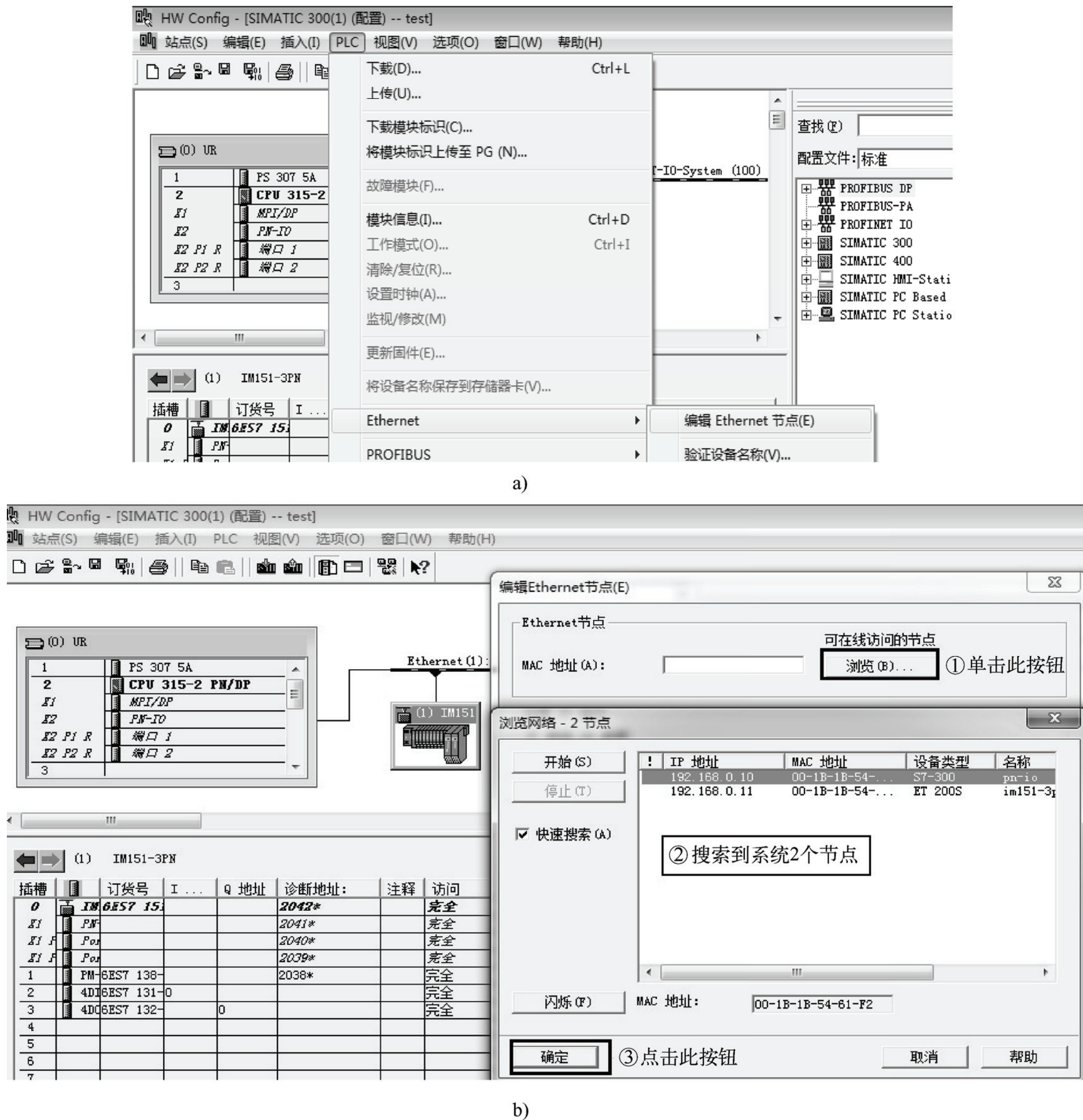


图 9-71　硬件组态和程序的下载

7. 系统调试

（1）查看系统网络结构是否正常

方法 1：进入硬件组态界面，查看网络节点是否连接，如图 9-72 所示，如果全部节点都能收到，说明网络正常连接。



a)

b)

图 9-72　查看系统网络结构是否正常

a) 查看网络节点路径　b) 网络节点全部搜索完成

方法 2：进入硬件配置界面→单击“组态网络”，出现图 9-73 所示界面。

（2）在线监控 PLC 变量

可以在线监控 OB1 程序，也可建立变量表 VAT_1 观察程序中变量的变化。例如，打开变量表输入需要监控的变量，并在线监控，如图 9-74 所示。

从图 9-74 中可以看到，远程 I/O 模块的 I0.0 变量外部开关接通，状态为“1”；由于 PLC 内部变量 MB1 未赋值，故远程输出模块状态都为“0”。在线修改 MB1 值为“16#03”，并“激活修改数值”，变量表如图 9-75 所示，观察远程 4DO 模块，输出状态发生变化；远程 I/O 模块工作状态如图 9-76 所示。

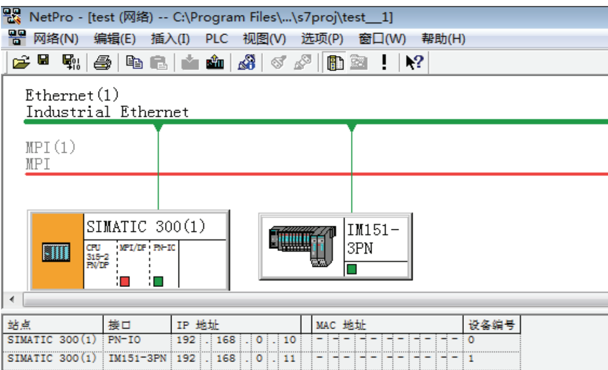


图 9-73　组态网络界面



图 9-74　在线监控变量表



图 9-75　PLC 对 MB1 赋值



图 9-76　ET200S 模块的现场运行状态

第三篇　自动控制系统集成

第 10 章　控制系统集成及其方法

10.1　系统集成的概念

系统集成（System Integration）是 20 世纪 90 年代在计算机业界用得比较普遍的一个词，包括计算机软件、硬件及网络系统的集成，以及围绕集成系统的相关咨询、服务和技术支持等。实际上集成的思想并不只在计算机业界专有，在传统制造业中，比如汽车工业，从手工作坊发展到大规模自动化生产方式后，为追求产品的批量和低成本，采用标准化生产线及加工工艺，零部件制造商专业化、标准化，总装厂与协作厂之间的协作生产化，都体现着系统集成的思想。

系统集成可以理解为按系统整体性原则，将原来没有联系或联系不紧密的元素组合起来，使其成为具有一定功能的、满足一定目标的、相互联系、彼此协调工作的新系统的过程。通过系统集成，可实现最大限度地提高系统的有机构成、系统的效率、系统的完整性、系统的灵活性，同时简化系统的复杂性，并最终为企业提供一套切实可行的、完整的解决方案。

系统集成可以是人员的集成、管理上的集成以及企业内部组织的集成，也可以是各种技术上的集成、信息的集成以及功能的集成等，因此系统集成涉及的内容非常广泛，其实现的关键在于解决系统之间的互连和相互间操作性的问题，它是一个多厂商、多协议和面向各种应用的体系结构。

自动控制系统的集成一方面可以实现企业管控一体化，提高企业管理信息化水平；另一方面可以实现生产设备的集中管理、分散控制，便于系统的远程监控和维护等，从而实现各种适应实际生产需要的控制功能。

现代控制系统的集成不再只是当初纯粹的集成电路控制模式，而是计算机技术与网络通信技术的紧密结合，近几十年经历了从集散控制系统（DCS）到现场总线控制系统（FCS）的发展。现场总线是一种应用于生产现场，在现场设备之间、现场设备与控制装置之间实行双向、串行、多节点数字通信的技术。现场总线作为智能设备的纽带，将挂接在总线上、作为网络节点的智能设备相互连接，构成相互沟通信息、共同完成自动控制功能的网络系统与控制系统。如果说计算机网络把人类引入到信息时代，那么现场总线则使自控系统与设备加入到信息网络的行列，成为企业信息网络的底层，使企业信息沟通的范围一直延伸到生产现场。但由于现场总线种类的多样，使得在短时间内现场总线的标准难以统一，设备的互连、互通与互操作问题很难解决，因此，在多标准、多厂商、多协议总线共存的情况下，解决好系统集成和应用集成的问题显得尤为重要。

现场总线控制系统的集成广泛采用现场总线技术、局域网技术和因特网技术。通过不断的发展，逐步实现了各种物理设备的互联技术即“设备集成”；实现了各种数据与信息的集成技术即“信息集成”，从而满足企业综合自动化的要求。

1．设备集成

设备集成是通过把设备的功能映射到在上位机运行的各种操作软件或工程工具中实现的。在控制系统中，系统中所有的智能仪表或设备的数据和功能无缝地为整个自动化系统所利用，使得最终用户接受在工业自动化和过程控制中存在多协议、多标准的现实。

设备集成可采用多种技术实现，如使用综合布线技术、通信技术、多媒体应用技术、安全防范技术以及网络安全技术等，可实现多个系统共用一个通信平台，实现系统各组成部分之间信息的正常交换，从而实现整个自动化系统的控制任务。

设备集成可能会涉及各种不同的设备，例如可以是计算机、PLC 等高端设备，也可以是阀门、开关等低端设备。选择集成设备时要注意在统一标准的前提下，选择性价比高、适用性好的产品构建网络平台或通信平台，使得系统具有共享软件、硬件和数据资源的能力，且具有对共享数据资源集中处理、管理和维护的能力。

2．信息集成

信息集成是指不同系统之间的信息能够实现共享，以满足信息交换的需求，或综合多个系统提供的信息以便作出决策。

信息集成是解决目前普遍存在的“信息孤岛”问题的重要方法，屏蔽了各种异构资源间的差异，使系统中各子系统和用户对交换的信息采用统一的标准、规范和编码，实现全系统信息共享，进而可实现用户软件间的交互和有序工作。例如在计算机集成制造系统（CIMS）中，计算机辅助设计（CAD）、计算机辅助制造（CAM）、计算机辅助工程（CAE）等单项技术都是生产作业上的“自动化孤岛”，单纯地追求每一单项技术上的最优化，不一定能够达到企业的目标；可以将各自的数据按某种标准规定的格式进行转换，得到一种统一的公用格式，便于经过数据库管理系统（DBMS）存入数据库以及检索利用。数据共享通常是信息集成的主要目标，选择恰当的 DBMS，组织好原始数据的提炼、加工和不断更新，是实现这一目标的重要工作。

3．集成系统的应用开发

集成系统的应用开发也是系统集成的一项重要任务。通过开发各种应用软件，保证其在信息应用系统中正常运行、不断完善和便于管理。目前为应用软件已能提供许多便利的、高效的开发工具和环境，恰当地选择或改善工具和环境，将可达到事半功倍的效果。

简言之，系统集成就是根据用户需求，择优选择各种技术和产品，将各类设备、各个分离的子系统连接成为一个完整、可靠、经济和有效的整体，使得各部分能彼此协调工作，发挥系统功能，达到整体优化和提高效益的目的。自动控制系统集成技术正在迅速发展，为控制系统的广泛应用和优化控制提供了有力的技术支持。

10.2 控制网络

控制网络用于完成各种数据采集和自动控制任务，是一种特殊的、开放的计算机网络，是工业企业综合自动化的基础。从现场控制网络节点的设备类型、传输信息的种类、网络所

执行的任务以及网络所处的环境等方面来看，都有别于其他计算机构成的数据网络。

自动控制网络可以通过网络互联技术实现不同网段之间的网络连接与数据交换，包括在不同传输介质、不同传输速率以及不同通信协议的网络之间实现互联，从而更好地实现现场信号检测、信息采集、控制和执行以及信息的传输、交换、存储与利用的一体化，以满足用户的需求。

10.2.1　网络节点

自动控制系统的网络节点通常分散在生产现场，大多是具有计算与通信能力的智能测控设备。它们可能是普通计算机网络中的 PC 或其他种类的计算机、操作站等设备；也可能是具有嵌入式 CPU，但功能比较单一、计算或其他能力远不及普通计算机，且没有键盘、显示等人机交互接口；也有的设备甚至不带有 CPU，只带有简单的通信接口。

例如具有通信能力的现场设备有条形阅读码、各类智能开关、可编程序控制器、监控计算机、智能调节阀、变频器和机器人等，它们都可以作为自动控制系统网络的节点使用。由于受到制造成本等因素的影响，作为控制网络节点的设备，在计算能力等方面一般比不上普通的计算机。

自动控制网络就是把单个分散的、有通信能力的测控设备作为网络节点，按照总线型、星型、树型等网络拓扑结构连接而成的网络系统。如图 10-1 所示的总线型控制网络连接，使得各个节点之间可以相互沟通信息、共同配合完成系统的控制任务。

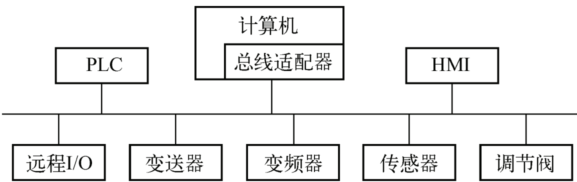


图 10-1　现场总线控制网络连接示意图

10.2.2　现场总线控制网络的任务

现场总线控制网络主要完成以下任务：

1）将控制系统中现场运行的各种信息传送到控制室，使现场设备始终处于远程监视之中。例如在控制室监视生产现场阀门的开度、开关的状态、运行参数的测量值及现场仪表的工作状况等信息。

2）控制室将各种控制、维护、参数修改等命令送往位于生产现场的测量控制设备中，使得生产现场的设备处于可控状态之中。

3）与操作终端、上层管理网络实现数据传输与信息共享。

此外，现场总线控制网络还要面临工业生产的高温高压、强电磁干扰、各种机械振动及其他恶劣工作环境，因此要求现场总线控制网络能适应各种可能的工作环境。

由于现场总线控制网络要完成的工作任务和所处的工作环境，使得它具有许多不同于普通计算机网络的特点。

影响控制网络性能的主要因素有网络的拓扑结构、传输介质的种类与特性、介质访问方式、信号传输方式以及网络监控系统等。为了适应和满足自动控制任务的需求，在开发控制

网络技术及设计现场总线控制网络系统时，应该着重于满足控制的实时性、可靠性以及工业环境下的抗干扰性等控制要求。

10.3 总线控制系统集成

10.3.1 现场总线控制系统集成框架

现场总线控制系统是以现场总线为通信介质的工业控制网络，是工业企业综合自动化的基础。根据计算机集成制造技术 CIMS 的观点，统一的集成现场总线控制系统结构可分为三个层次，如图 10-2 所示，由低到高依次是现场控制层、过程监控层和企业管理层。

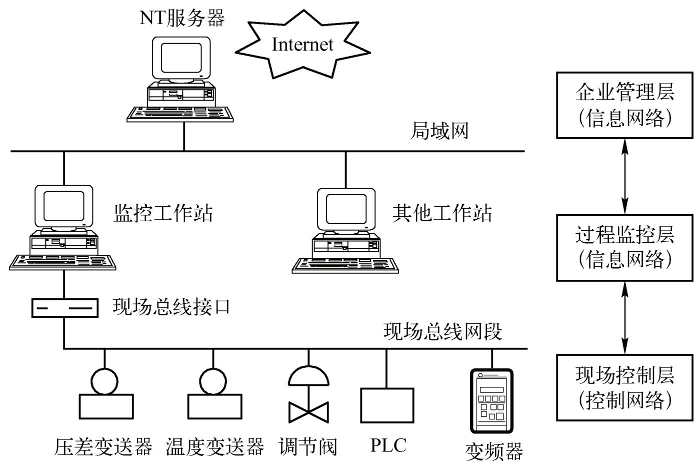


图 10-2 现场总线网络集成层次图

1．现场控制层

现场控制层是集成系统的底层设备，由现场控制设备作为网络节点构成整个系统的控制网段，主要实现对生产过程的检测和控制，包括传感变送、PID 调节等功能的集成。可以包括 FF 的 H1 网段、PROFIBUS、HART、LonWorks 等网段，同种通信协议的网段间可采用网桥、中继器连接；不同通信协议的网段间可采用网关连接，也可直接在操作站的计算机内交换信息；带有 AI、AO、DI、DO 和 PID 功能块的现场设备负责完成生产过程的参数测量、数据传输与过程控制，即由现场总线网络完成现场控制层的自动化任务。

在现场控制层的应用集成中，标准功能块（Function Block，FB）与设备描述（Device Description，DD）是应用集成的基础。现场装置使用 FB 完成控制策略；DD 用来描述现场装置通信所需的所有信息，可以将不同供应商的设备添加到控制网络中，使现场装置实现真正的互操作性。

现场控制层对网络传输的数据吞吐量要求不高，但对通信响应的实时性和确定性要求较高。该层网络上传输的信息如果出现延时或丢失现象，就会降低控制系统的性能甚至使控制系统不稳定。

2．过程监控层

过程监控层一般由担任监控任务的工作站、PC 或控制器作为网络节点构成局域网段，主

要用于完成对控制系统的组态，执行对控制系统的监视、报警和维护，将采集到的现场信息置入实时数据库，进行优化计算和先进控制。该层网络可通过扩展槽网络接口板与现场总线相连，协调网络节点之间的数据通信；也可通过专门的现场总线接口实现现场总线网段与以太网段的连接。

过程监控层上传输的信息具有周期性、实时性的特点；如果该层网络上传输的实时信息出现较大延时或丢失时，就会导致多个设备不能协调工作。

3．企业管理层

企业管理层位于集成系统的最上层，由高性能计算机、工作站、PC 等网络节点构成，是实现企业信息集成和管、控一体的重要组成部分，主要用于企业的计划、销售、生产、库存以及企业经营等方面信息的传输，从而在分布式网络环境下构建一个安全的远程监控系统。企业管理层还可通过 Internet 实现与企业内远程网点（如商业网点）的信息集成。

在企业管理层上传输的信息一般都是非实时性的，并且数据包较大，信息传输频率较低；数据通信的发起是随机的、无规则的，数据吞吐量较大，因此要求网络必须具有较大的带宽。

在现场总线网络集成中，现场控制层是整个系统的核心，只有确保总线设备之间可靠、准确、完整的数据传输，上一层网络才能获取正确的信息以及实现监控、优化等功能。总之，要使一个企业内部实现信息控制一体化，组成一个协调的网络整体，现场总线控制网络和信息网络的集成是要解决的问题。基于开放的现场总线协议标准，为控制网络与信息网络的连接提供了方便，通过现场总线与信息网络互联，并进一步接入 Internet，从而实现企业内、外信息控制网络的全方位集成。

10.3.2　现场总线控制系统集成的原则

系统集成的本质是通过不同设备之间或不同子系统之间的互联、互通，实现信息资源的共享，以便更好地完成对整个系统的管理和决策。系统集成需要遵循以下几个原则：

1．开放性、标准化原则

采用的标准、技术、结构、系统组件、用户接口等遵从开放性和标准化的要求，即选择符合国家或国际标准的产品或技术，使得集成的系统具有兼容性和可移植性。

2．实用性和先进性原则

实用有效是最主要的设计目标，设计结果应能满足需求，且切实有效；设计上确保设计思想先进、网络结构合理、网络硬件设备先进、开发工具友好，使系统具有可延续性和可扩展性。

3．可靠性和安全性原则

稳定、可靠、安全地运行是系统集成的基本出发点。集成后的系统应具有更高的容错性和抗干扰性，使系统能在相应条件下安全运行，满足各项性能指标的要求，实现系统功能的最大化。

4．灵活性和可扩展性原则

系统集成配置灵活，提供备用方案和可选方案；能够根据要求在性能和规模等方面进行扩展，以适应系统发展的需要。

5．经济性原则

在满足系统需求的前提下，应尽可能选用价格便宜的设备，以便节省系统投资。

总之，一个高性能的自动控制系统，应能够方便地对系统的资源统一管理和调控，快速响应用户需求，更好地为用户提供信息服务。

10.4 现场总线控制系统和网络的集成方法

现场总线控制系统（FCS）是在传统的仪表控制系统和集散控制系统（DCS）的基础上，利用现场总线技术逐步发展而成的。目前工业中仍然使用大量的模拟仪表和集散控制系统，而集散控制系统本身也在不断地发展与完善，现场智能仪表不可能完全取代模拟仪表，FCS 也不可能完全取代 DCS。因此，FCS 和 DCS 集成是自动控制系统发展的必由之路，既考虑了现实情况，又有利于 FCS 的发展和推广；同时，随着 Intranet 和 Internet 的迅速发展，发展 FCS 和网络的集成技术也成为必然；目前现场总线处于多种总线并存的格局，也就是说，现场总线的标准是多元化的，发展多种现场总线之间的集成技术也是非常必要的。

可见，由于自动化技术发展的延续性和继承性、各种计算机主流技术在工业控制领域的渗透和应用以及各种现场总线技术的发展和竞争，现场总线控制系统集成技术的研究和发展面临着多系统集成、多技术集成和多总线集成的局面，其系统集成方法可分为以下三类：

1）FCS 和 DCS 的集成方法。

2）FCS 和网络的集成方法。

3）FCS 和其他现场总线的集成方法。

10.4.1 FCS 和 DCS 的集成方法

DCS 技术的发展比较成熟，已广泛应用于生产过程自动化控制领域。FCS 的发展需要借助于 DCS，这样既能丰富 DCS 的功能，又能推动现场总线及其控制系统的发展。FCS 和 DCS 的集成方式可以有三种：现场总线与 DCS 输入/输出总线的集成、现场总线与 DCS 网络层的集成、FCS 与 DCS 的集成。

1．现场总线与 DCS 输入/输出总线的集成

DCS 的控制站主要由控制单元和输入/输出单元组成，这两个单元之间通过 I/O 总线连接。其主要功能有：

1）通过 I/O 总线与输入/输出单元通信，建立 I/O 数据库。

2）实现运算和控制功能，完成用户组态的控制策略。

3）与 DCS 网络通信。

在输入/输出单元的 I/O 总线上挂接了各类 I/O 模板，常用的有模拟量输入、模拟量输出、数字量输入及数字量输出等模板，通过这些模板与生产过程建立 I/O 信号联系。

如图 10-3 所示，现场总线接口单元或现场总线接口板挂接在 DCS 的 I/O 总线上，现场总线的数据通过此接口映射为 DCS 的 I/O 总线可以接受的相应数据信息， 使得在 DCS 控制器所看到的现场总线发来的信息就如同来自一个传统的 DCS 设备一样，这样便实现了现场总线与 DCS 输入/输出总线的集成，即现场总线和 DCS 控制站的集成。

现场总线和 DCS I/O 总线的集成具有以下特点：

1）只需安装现场总线接口设备，无须改变或升级 DCS。

2）可以充分利用 DCS 控制站的运算和控制功能块。由于初期开发的现场总线仪表中的功能块数量和种类有限，这样就可以利用比较完善的 DCS 的功能块资源。

3）利用已有 DCS 的技术和资源，不仅投资少、见效快，而且便于推广现场总线的应用。

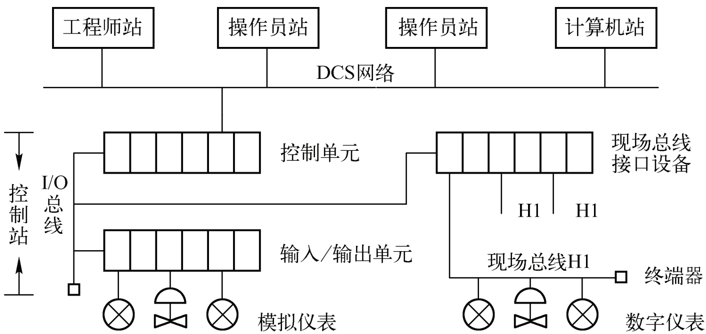


图 10-3　现场总线和 DCS I/O 总线的集成

这种集成方案可用于 DCS 系统已经安装并稳定运行，而用现场总线对原有系统的控制进行小容量扩充的场合以及现场总线技术的初期应用中。

2．现场总线与 DCS 网络层的集成

在 DCS 控制站的 I/O 总线上集成现场总线是一种最基本的初级集成技术。如图 10-4 所示的现场总线接口设备不是挂在 DCS 的 I/O 总线上，而是挂在 DCS 的网络上，即现场总线与 DCS 的集成还可以在 DCS 的网络层上实现。其中，现场总线服务器是一台完整的计算机，在其内安装了现场总线接口卡和 DCS 网络接口卡；现场设备通过现场总线与其接口卡通信；而现场设备中的输入、输出、控制和运算等功能块可以在现场总线上独立构成控制回路，不必借用 DCS 控制站的功能。

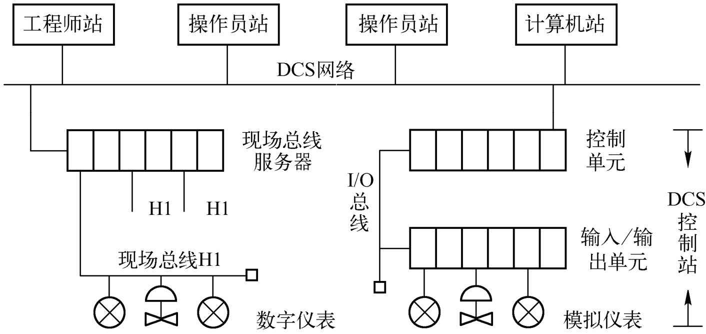


图 10-4　现场总线与 DCS 网络层的集成

现场总线服务器通过其 DCS 网络接口卡与 DCS 网络通信，可以把现场总线服务器看作 DCS 网络上的一个节点或 DCS 的一台设备。现场总线服务器和 DCS 之间可互相共享资源；现场总线服务器可以不配操作员站或工程师站，而直接借用 DCS 的操作员站或工程师站。

现场总线与 DCS 网络层的集成具有以下特点：

1）除了安装现场总线服务器外，不需要对 DCS 控制站进行变动。

2）在现场总线上可以独立构成控制回路，实现彻底的分散控制。

3）现场总线服务器中有一些高级功能块，可以与现场仪表中的基本功能块统一组态，构成复杂控制回路。

4）DCS 可以访问现场设备中更多的信息。

5）利用已有 DCS 资源，不但投资少、见效快，而且便于推广现场总线的应用。

3．FCS 与 DCS 的集成

在上述两种集成方式中，现场总线不是独立的，需要借用 DCS 的某些资源。随着现场总线技术的不断发展，FCS 已成为一种独立的、完整的控制系统，因此现场总线与 DCS 之间的集成变成了两个独立系统之间的集成。

FCS 和 DCS 的集成方式有两种，一种是 FCS 和 DCS 分别挂接在 Intranet 上，通过 Intranet 间接交换信息；另一种是 FCS 网络通过网关与 DCS 网络集成，在各自网络上直接交换信息。

如图 10-5 所示，FCS 通过网关与 DCS 集成，网关用来完成 FCS 与 DCS 网络之间的信息传递，同时整个集成系统还可以通过 Web 服务器实现 Intranet 与 Internet 的互联。

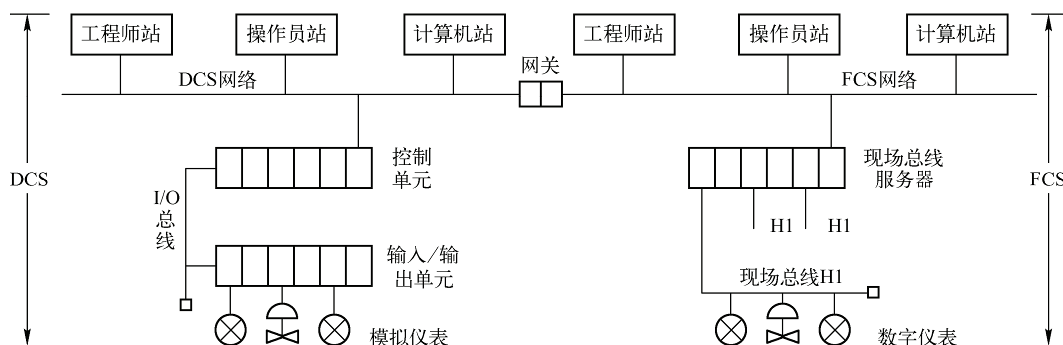


图 10-5　FCS 通过网关与 DCS 集成

FCS 和 DCS 的集成具有以下特点：

1）FCS 是一个完整的系统。

2）FCS 和 DCS 可分别安装，对两种系统不需做任何变动，只需在两种系统之间安装网关即可。

3）能有效保护用户对 DCS 的先期投资，同时还有利于 FCS 的发展和推广。

这种集成方式通过专门设计的网关接口实现了 FCS 和 DCS 的双向连接，即 DCS 操作员站能访问现场设备中的所有信息，而 FCS 也能获取 DCS 的各种信息，因此便于实现 FCS 和 DCS 的协调控制。但对于网关接口技术要求较高，此接口需要完成双向的协议转换，为 FCS 和 DCS 提供透明的数据访问。这种集成方式主要适用于规模较大的控制系统。

10.4.2　FCS 和网络的集成方法

FCS 是一种分布式网络控制系统，它的基础是现场总线，且位于网络结构的最底层，因而被称为底层网（Infranet）。FCS 的上层是 Intranet，Intranet 下面可以挂接多个 FCS 或 DCS

的底层网或控制网络；Intranet 的上层网是 Internet，Internet 下面可以挂接多个 Intranet。FCS 和网络的集成可以通过 FCS 和 Intranet 的集成、FCS 和 Internet 的集成来实现。

1．FCS 和 Intranet 的集成

高度分散的工业现场传感器、变送器以及执行器等智能设备通过现场总线网络连接到控制器或上位管理机上，构成局域网络控制系统。这种系统可以节省大量的传输导线，增强整个系统的扩展性，具有较长的传输距离和较强的抗干扰能力，可实现无上位机的全分布式无主工作，为工业控制和企业管理决策提供了一种全新的解决方案，大大促进了控制技术的发展和信息系统的集成。

FCS 和 Intranet 的集成可以通过以下几种方法来实现。

（1）通过网桥、中继器等转换接口来实现

这种集成方式需要通过硬件来实现。在底层网段与中间监控层之间加入中继器、网桥以及路由器等专门的硬件设备，使底层控制网络与上层信息网络集成。硬件设备可以是一台专门的计算机，依靠其中运行的软件完成数据包的识别、解释和转换；对于多网段的应用，它还可以在不同网段之间存储转发数据包，起到网桥的作用。

通过转换接口实现集成的方式功能较强，但实时性较差。信息网络一般采用 TCP/IP 的以太网，而 TCP/IP 没有考虑数据传输的实时性，当现场设备有大量信息上传或远程监控操作频繁时，转换接口将成为实时通信的瓶颈。

（2）采用 DDE 技术

当 FCS 和 Intranet 之间具有中间系统或共享存储器工作站时，可以采用动态数据交换（Dynamic Data Exchange，DDE）方式实现两者的集成，其实质是各应用程序通过共享内存来交换信息，中间系统中的信息处理既是现场总线控制网络的工作站，也是 Intranet 中的工作站。其中运行两个程序：一是为 Intranet 数据库提供实时数据信息的通信程序；另一个是数据访问应用程序接口，它接收 DDE 服务器实时数据，并写入数据库服务器中，供信息网络实现信息处理、统计分析等功能。

DDE 方式使用灵活、具有较强的实时性，且比较容易实现，可以采用标准的 Windows 技术。但是 DDE 的速度是主要问题，适合配置简单的小系统。

（3）采用 OPC 技术

OPC（OLE for Process Control）是用于过程控制的对象链接与嵌入技术。它是由世界上多个自动化公司、软硬件供应商与微软合作开发的一套数据交换接口的工业标准，能够为现场设备、自动控制应用、企业管理应用软件之间提供开放的、一致的接口规范，为来自不同供应商的软硬件提供“即插即用”的连接。OPC 采用客户/服务器（Client/Server）结构，服务器对下层设备提供标准的接口，使得现场设备的各种信息能够进入 OPC 服务器，从而实现向下互联；服务器对上层设备提供标准的接口，使得上层 Intranet 设备能够取得 OPC 服务器中的数据，从而实现向上互联，如图 10-6 所示。由于 OPC 技术的“即插即用”，设备可以很容易加入现有系统并立即使

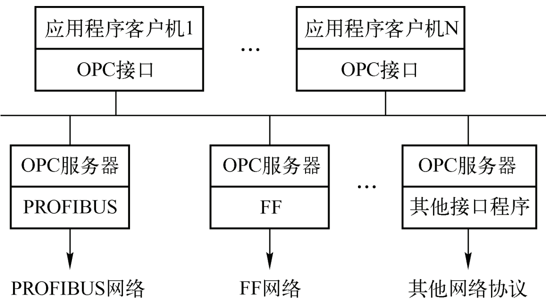


图 10-6　采用 OPC 技术的系统集成

用，不需要复杂的配置，也不会影响现有系统。

（4）使用 ODBC 技术

ODBC（Open Database Connectivity）是开放数据库互联技术，是一种用来在相关或不相关的数据库管理系统（Database Management System，DBMS）中存取数据的标准应用程序编程接口（Application Programming Interface，API），是为用户提供简单、标准、透明、统一的数据库联接的公共编程接口，在各个厂家的支持下能为用户提供一致的应用开发界面，使应用程序独立于数据库产品，实现各种数据库之间的通信。

实时数据库与关系数据库是企业信息集成的重要支撑，是实现数据交换与资源共享的有效工具。ODBC 技术可作为连接数据库的统一界面标准，通过它能创建与各种数据库进行交互的程序，是系统集成的又一有效工具。

（5）采用统一的协议标准

FCS 和 Intranet 网络采用统一的协议标准将成为现场总线控制网络和 Intranet 集成的最终解决方案。由于控制网络和信息网络采用了面向不同应用的协议标准，两者集成需要进行数据格式的转换，这样既不能确保数据完整，又使得系统复杂化。但如果信息网络能提高其实时性，控制网络能提高其传输速率，则两者的兼容性就会提高；如果从现场控制层到远程监控层都使用统一的协议标准，则可以确保信息准确、快速、完整的传输，并能简化系统设计。发展成熟后的工业以太网就是这种最终解决方案的产物。

目前，像 FF、PROFIBUS 和 WorldFIP 等现场总线致力于使自己的通信协议尽量兼容 TCP/IP 协议，以便实现与以太网和 Intranet 的集成，最终实现统一的网络结构。

2．FCS 和 Internet 的集成

在现代社会中，网络已把家庭、企业和社会连接在一起，而 Internet 也把整个世界紧密地联系起来。如果把控制网络中的实时控制信息和数据网络中管理决策信息结合起来，会使网络功能得到更大的发挥。

实现控制网络与信息网络的紧密集成是建立企业综合实时信息库的基础，可以保证系统数据的一致性、完整性和互操作性，能够为企业的优化控制、调度决策提供依据。

FCS 和 Internet 的集成有两种方式，一种是 FCS 通过 Intranet 间接和 Internet 集成；另一种是 FCS 直接和 Internet 集成。

FCS 和 Intranet、 Internet 网络的集成可以构成远程监控系统，使用户通过信息网络中标准的图形界面实时掌握生产设备的运行状况、实现远程监控和诊断维护等功能，同时在完成内部管理的同时，还可以加强与外部信息的交流。

10.4.3　FCS 和其他现场总线的集成方法

目前现场总线种类众多，仅 IEC 通过的现场总线标准 IEC61158 就包含 20 种类型。另外，还有其他国家的标准现场总线。现场总线是 FCS 的基础，怎样将多种现场总线集成在一个系统中，共同配合完成工厂的测控任务，也是 FCS 开发者面临的一个重要问题。

不同现场总线系统之间的集成需要解决各种信息映射与协议转换、设备之间的互操作性以及各种现场总线设备的组态、监控与操作统一等关键技术。总的来说，不同现场总线系统之间的集成可以采用两种方式：一种是通过网关给各个现场总线之间提供转换接口；另一种是给各个现场总线提供标准的 OPC 接口。采用网关的集成方法开发工作量大，由于不同现场

总线均有自己的协议，不可能开放任意两种总线之间的网关，因此不具有通用性；采用 OPC 接口的集成方法开发工作量小，具有通用性。

虽然利用网关对协议进行转换识别既增加集成系统的硬件投入，又增加了网络延迟，但对于多种总线并存的现状，可以将其视为一种过渡时期采取的措施。现场总线标准的统一工作虽然在短期内还难以实现，有很多问题还亟待解决，但随着现场总线技术的不断发展和完善，不同制造商的总线模块产品会走向互相兼容、互相统一的时代，到那时，现场总线无疑会进入到一个发展更加迅猛的全新阶段。

第 11 章　S7-300 PLC 与 S7-200 PLC 的系统集成

11.1　基于 PROFIBUS-DP 的系统构建与运行

11.1.1　控制要求及硬件配置

采用 PROFIBUS-DP 通信方式实现 S7-300 PLC 和 S7-200 PLC 之间的数据通信。

分析：S7-300 PLC 基于模块化结构设计，适用于中小型控制系统。主站选择 S7-300 CPU315-2PN/DP。CPU315-2PN/DP 为中央处理单元，工作电源 DC24V，带有 384KB 工作存储器、1 个 MPI/DP 12Mbit/s 接口、2 个以太网 PROFINET 接口、双端口交换机，使用时需要微型存储卡（简称为 MMC）；具有中等规模程序量，可在 SIMATIC S7-300 系统中用作 PROFINET IO 控制器以及标准 PROFIBUS-DP 主站，还可用作分布式智能从站（DP 从站）。硬件组态、参数配置及程序的编写由 STEP 7 V5.5 软件完成。

S7-200 PLC 不支持 DP 通信协议，自身也不带 PROFIBUS-DP 接口，不能直接作从站，但可以通过添加 EM277 总线模块，将 S7-200 PLC 作为从站连接到 PROFIBUS-DP 网络中。从站选择 S7-200 CPU226 整体式 PLC，与 EM277 模块配合使用。通过 EM277 模块可将 S7-200 CPU226 连接到 PROFIBUS-DP 网络上；S7-200 PLC 的通信编程采用 STEP Micro/WIN 4.0 SP5 软件完成；在通信中需要使用输入/输出缓冲区，该缓冲区被定义在 S7-200 PLC 的变量存储器 V 中。

如图 11-1 所示，EM277 通过串行 I/O 总线连接到 CPU226 PLC 的扩展口上，S7-300 CPU315-2PN/DP 上的 DP 接口通过 PROFIBUS-DP 总线与 EM277 模块相连构成 PROFIBUS 网络，其中 S7-300 PLC 作为系统的主站，完成网络组建功能。

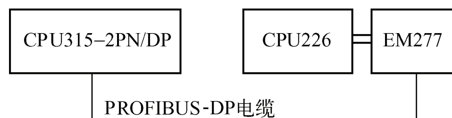


图 11-1　系统结构图

EM277 模块是智能模块，其 RS-485 接口是隔离型的，端口波特率为 9.6kbit/s~12Mbit/s，能自适应系统的通信速率；作为 DP 从站，EM277 接受来自主站的 I/O 组态，向主站发送和接收数据；主站可以读写 S7-200 PLC 的 V 存储区，每次可以与 EM277 交换 1~128 个字节。

11.1.2　从站的设置

PROFIBUS-DP 的所有组态工作由主站完成，不需要在 S7-200 PLC 一侧对 PROFIBUS-DP 通信组态和编程，只需将要进行通信的数据整理并存放在主站组态时为其指定的 V 存储区地

址里就可以了。在从站方面，主要是设置从站设备的硬件地址，该地址必须与主站中的组态地址相匹配。具体设置步骤如下：

1）关闭模块的电源。

2）在 EM277 模块上设置已经定义的 PROFIBUS-DP 地址，如图 11−2 所示，假设在主站组态时将从站模块地址设为“5”，旋转地址开关对应的个位、十位箭头，使箭头指向从站所需的站号“5”。

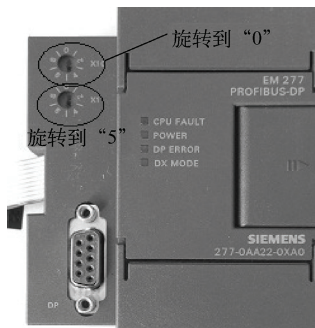


图 11−2　设置从站地址

3）打开模块电源。只有在重新打开电源之后，系统才能识别新设置的 PROFIBUS-DP 从站地址。

11.1.3　主站的硬件组态

1．有关硬件组态

运行 PROFIBUS 系统之前，需要先对系统及各站点进行硬件配置和相关参数设置，即对系统进行硬件组态。这项工作可以由 STEP 7 编程软件实现，该软件集成了 PROFIBUS 系统中主要设备的所有 PROFIBUS 通信功能，可完成以下工作：

1）在 STEP 7 编程软件中生成一个与实际的硬件系统完全相同的系统，包括生成机架和模块、CPU 型号/参数设置、网络参数配置、远程从站硬件配置、模块的地址分配、主-从站进行数据传输时的输入/输出字（或字节）数及通信映像区地址、系统故障模式设定等内容。

2）设置系统诊断。通过设置系统诊断，可以实现在线检测系统并找到故障点，读到故障的提示信息，通过两种方式进行信息显示：

① 快速浏览 CPU 的数据和用户编写的程序在运行中的故障原因。

② 用图形方式显示硬件配置。例如显示模块的一般信息和模块的状态、显示模块故障、显示诊断缓冲区的信息等。

CPU 还可以显示循环周期、已占用和未占用的存储区、通信的容量和利用率及显示性能数据等众多信息。

3）第三方设备集成及 GSD 文件。当 PROFIBUS 系统中需要使用第三方设备时，应该得到设备厂商提供的 GSD 文件。将 GSD 文件复制到 STEP 7 或 COM PROFIBUS 软件指定目录下，使用 STEP 7 或 COM PROFIBUS 软件可在友好的界面指导下完成第三方产品在系统中的配置及参数设置等工作。

2．硬件组态的操作过程

下面对 CPU315-2PN/DP 和 CPU226 通信模块 EM277 构成的系统进行硬件配置和参数设置，然后在主站和从站之间组态通信数据。

1）进入 SIMATIC Manager 界面，单击“文件”→“新建”菜单项，创建新项目“S7200-DP”，在新项目中插入一个 CPU315-2PN/DP 站点，如图 11-3 所示。

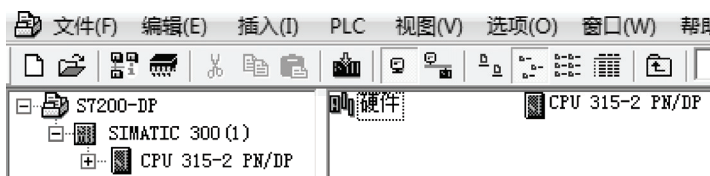


图 11-3　创建一个新项目

2）打开“HW Config”编辑器，在界面右侧的硬件目录中，依次查找并插入机架、CPU315-2PN/DP PLC，如图 11-4 所示；设置“MPI/DP”对象属性，如图 11-5 所示，建立 PROFIBUS 网络，以便将 CPU 连接到 PROFIBUS 网络中。

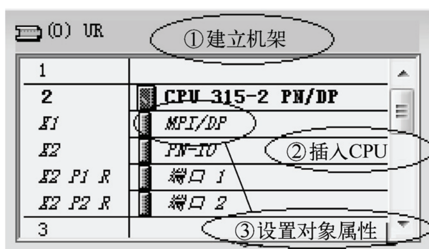


图 11-4　硬件配置



图 11-5　网络连接

3）导入 EM277 模块的 GSD 文件。如图 11-6 所示，在硬件组态界面，单击菜单“选项”→“安装 GSD 文件…”，弹出图 11-7 所示对话框。

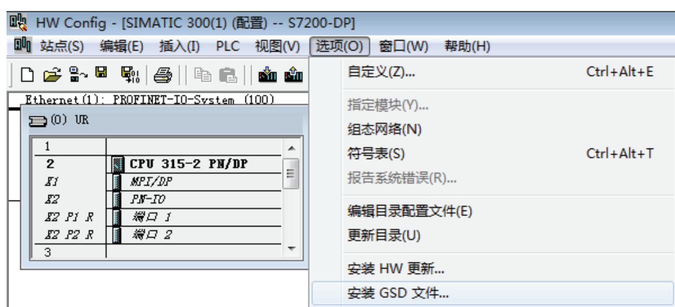


图 11-6　安装 GSD 文件

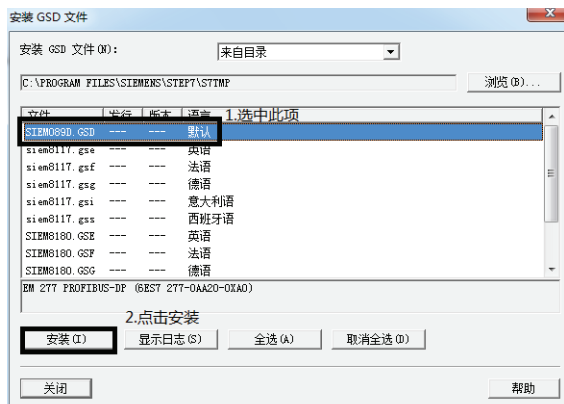


图 11-7　导入 EM277 的 GSD 文件

EM277 模块的 GSD 文件名为“SIEM089D.GSD”。STEP 7 编程软件在默认情况下，硬件目录中不包含此硬件，可以通过西门子官网或其他 SIMATIC 客户支持网站下载；通过 GSD 文件将 EM277 集成到 STEP 7 软件的硬件目录下，以便下一步对从站进行组态。

4）如果 GSD 文件安装成功，则在“HW Config”界面的右侧设备“目录”中可以找到 EM277 模块信息。进入 EM277 模块路径，如图 11-8 操作步骤①所示，将 EM277 模块添加到机架上的 PROFIBUS-DP 网段，然后设置该从站地址，地址要与实际 EM277 上的拨位开关设定的地址值一致（根据前面 EM277 上的拨位开关设定的从站地址，组态地址设为“5”；根据系统需要的通信字节数，选择通信方式，例如操作步骤②中选择“2 字节入/2 字节出”的数据交换方式。图 11-9 所示为配置完成的系统硬件组态界面。



图 11-8　选择 EM277 的通信方式

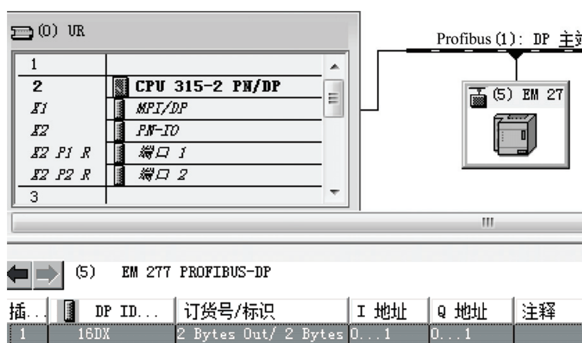


图 11-9　配置 EM277 网络信息

5）定义通信变量地址。要完成主站和从站之间的数据通信，还需要在通信两端为接收和发送数据定义变量存储区地址。本系统接收和发送数据各定义了 2 个字节长度，由图 11-9 可见，S7-300 PLC 的数据接收区地址为 IB0、IB1，发送区地址为 QB0、QB1；如果设定 S7-200 PLC 的通信缓冲区起始地址为 VB100，则 CPU226 缓冲区地址设置操作如图 11-10 所示，数据接收区地址为 VB100、VB101，发送区地址为 VB102、VB103，系统数据交换情况如图 11-11 所示。

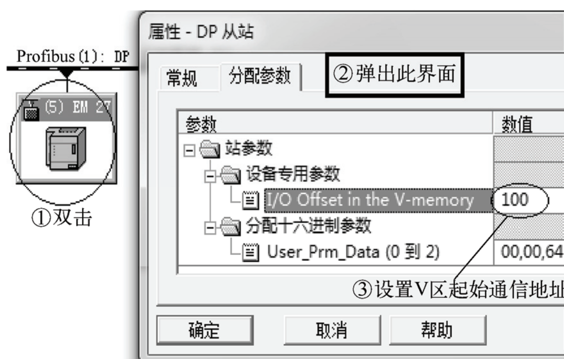


图 11-10　指定接收区的起始地址

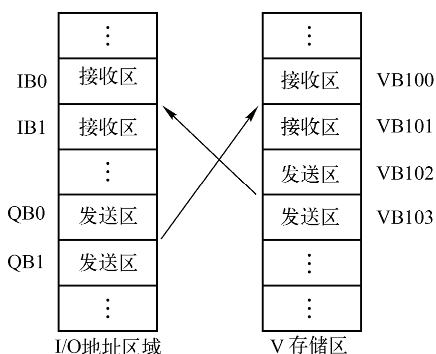


图 11-11　数据的交换

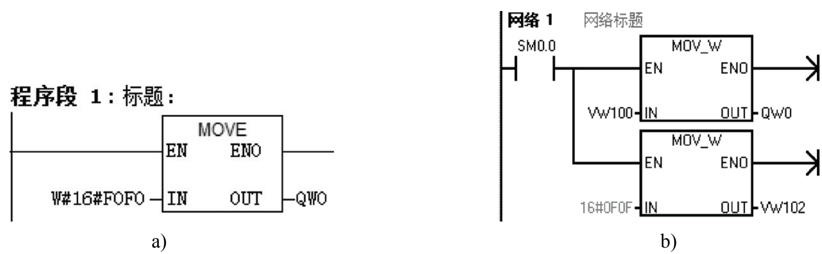
需要注意的是，在硬件组态时要保持已指定数据通信所使用的数据类型的一致性。所谓数据的一致性是指在 PROFIBUS-DP 数据传输时，数据的各个部分不会割裂开来传输，保证同时更新，即字节一致性保证字节作为整个单元传送，字一致性保证组成字的两个字节一起传送，缓冲区一致性保证数据的整个缓冲区作为一个整体一起传送。例如，对于四个数据字

的数据长度，如果使用了一个字的一致性，则将无法确保从站可以按照主站发送的顺序接收到这些数据字，只能保证单个字的正确接收。因此在组态时要注意数据一致性问题，以免数据传输错误。

6）保存并编译硬件组态文件，并将组态文件下载至 CPU315-2PN/DP PLC 中。

11.1.4　程序编写

主、从站程序设计如图 11-12 所示，含义为给主站通信输出变量 QW0、从站输出变量 VW102 赋值，以便系统数据交换和在线观察；分别将主、从站程序保存和下载至相应的 PLC 中。



a)　　b)

图11-12　主、从站程序设计

a) 主站程序　b) 从站程序

11.1.5　系统联调

将系统投入运行，分别观察主、从站变量变化情况，如图 11-13 所示。主站变量 IB0、IB1 接收来自从站 VB102、VB103 发送的值；从站变量 VB100、VB101 接收来自主站变量 QB0、QB1 发送的值。

VAT_1 -- @S7200-DP\SIMATIC 300(1)\CPU 315-

	地址		符号	显示格式	状态值
1	QB	0		HEX	B#16#F0
2	QB	1		HEX	B#16#F0
3	IB	0		HEX	B#16#0F
4	IB	1		HEX	B#16#0F

a)

状态表

	地址	格式	当前值	新值
1	VB100	十六进制	16#F0	
2	VB101	十六进制	16#F0	
3	VB102	十六进制	16#0F	
4	VB103	十六进制	16#0F	
5		有符号		

b)

图 11-13　主、从站变量变化情况

a) 主站变量观察图　b) 从站变量观察

在设计系统时需要注意：

1）在数据通信中，主、从站发送的数据分别存储在接收方的接收区中，必须将接收的数据“转移”到其他数据区，否则在下一次数据发送时这些数据将被新的数据覆盖。

2）在硬件组态中需要注意数据一致性问题，以免数据传输混乱。

11.2　基于 PROFINET 的系统构建与运行

11.2.1　控制要求及硬件配置

S7-200 PLC 可以通过 CP243-1 模块作为服务器或客户端与系统建立 S7 连接，下面案例

是将 S7-200 配置为客户端，由 S7-200 主动激活与远程服务器的 S7 连接。

硬件配置如下：CPU315-2PN/DP PLC、CPU224CN PLC 和 CP243-1 模块。

11.2.2　S7-200 PLC 的通信配置

在 STEP 7-Micro/WIN 中使用以太网向导为 S7-200 PLC 进行 S7 通信配置，配置步骤如下：

1）首先打开 STEP 7-Micro/WIN 软件，单击“工具”→“以太网向导”，如图 11-14 所示。

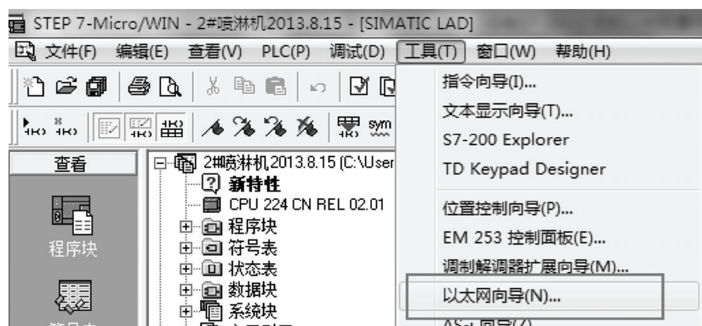


图 11-14　配置路径

2）如图 11-5 所示，“以太网向导”打开的界面中将有该向导的相关描述，单击“下一步”按钮进行相关组态。

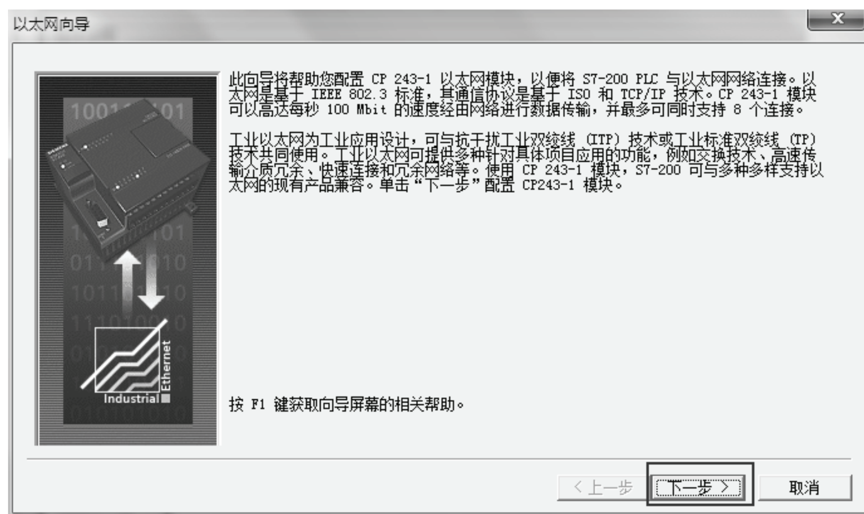


图 11-15　以太网向导介绍界面

3）如图 11-16 所示，可以通过单击“读取模块”按钮自动识别 CP243-1 以太网模块位置，也可以通过手动方式输入模块的位置，完成后单击“下一步”按钮。

4）如图 11-17 所示，为 CP243-1 模块分配一个单独的 IP 地址及相应的子网掩码，单击“下一步”按钮。

5）图 11-18 所示的对话框为 CP243-1 进行 S7 通信的连接数量的设置，通过 S7 连接可以与通信伙伴进行读写数据操作。模块命令字节中填写模块的输出地址，建议使用默认值。单击“下一步”继续进行 S7 连接组态。

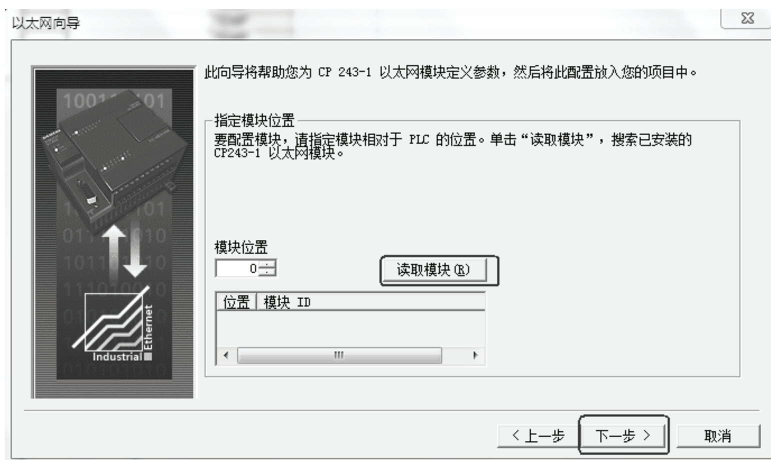


图 11-16　以太网模块位置

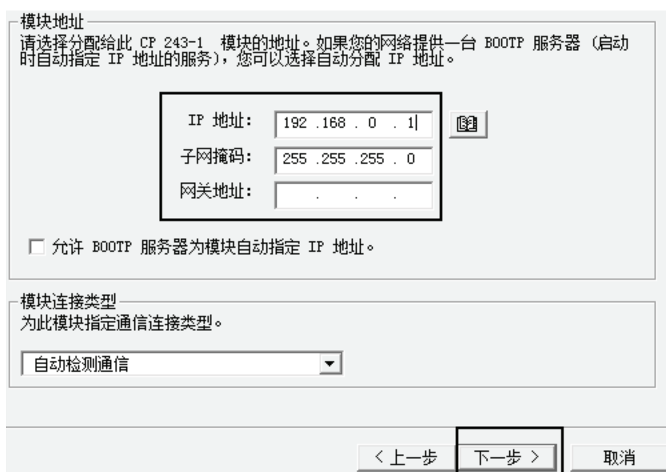


图 11-17　以太网模块地址

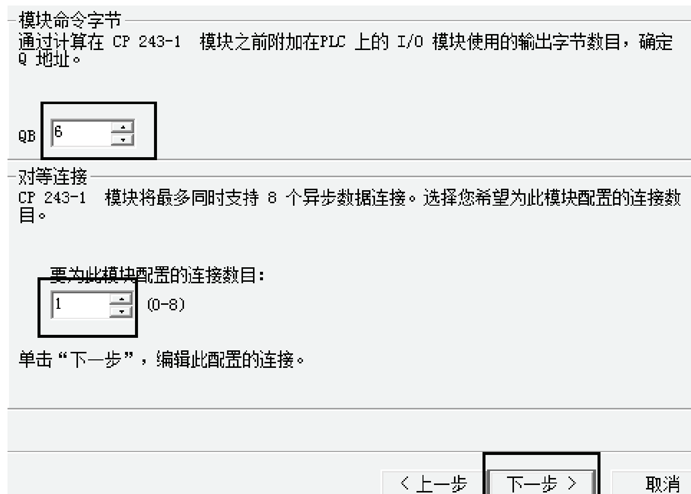


图 11-18　模块通信配置

6）如图 11-19 所示，本例中 S7 连接组态为客户端连接。S7-200 和 S7-300 的 S7 连接通过 TSAP 来定义，S7-200 侧的客户端连接为单边组态，远程 TSAP 设置为 03.02，该 TSAP 含义如下：

03：单边组态连接。

02：S7-300 站中 PLC 的槽号。

配置之后单击“数据传输”按钮。

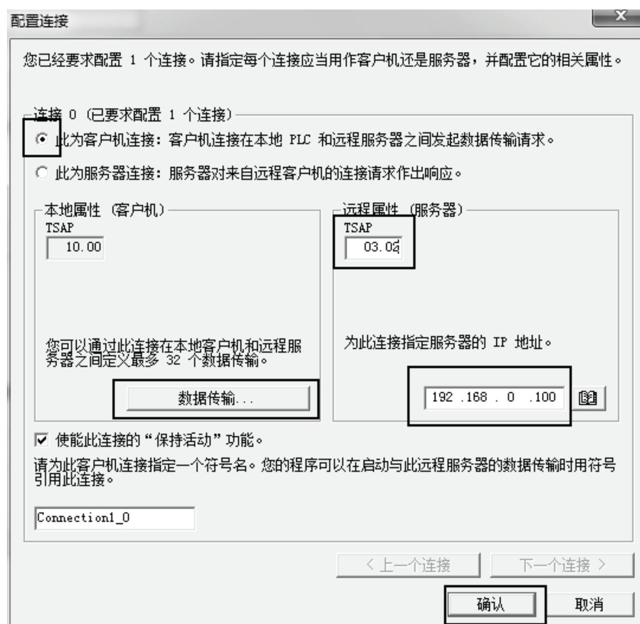


图 11-19 配置连接

7）如图 11-20 所示，如果没有定义数据传输，则单击“新传输”按钮进行设置。

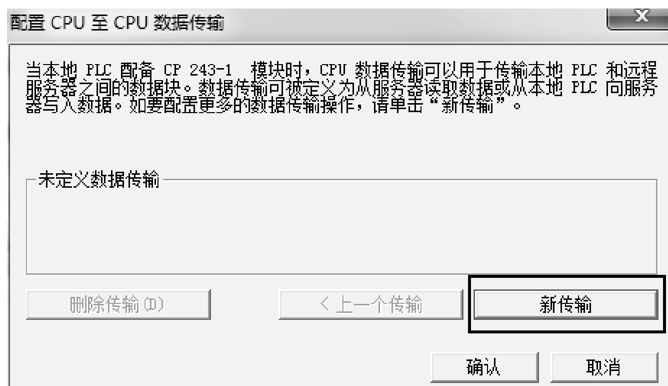


图 11-20 配置数据传输

8）如图 11-21 所示，进行数据传输方向、数据大小和数据存储地址设置。单击“下一个传输”进行写入 S7-300 PLC 数据设置界面。

9）如图 11-22 所示，从客户端写入数据到远程服务器 S7-300 PLC 中，配置完成后单击“确认”按钮，弹出图 11-23 所示的界面。

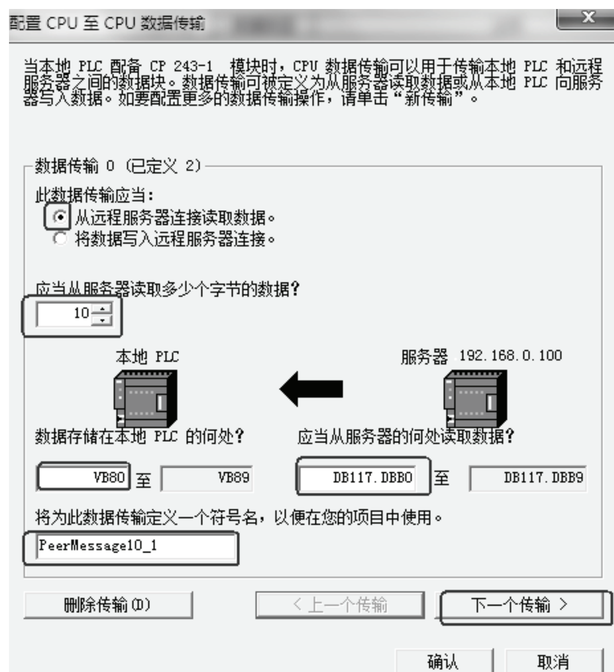


图 11-21　S7-200 PLC 读取数据

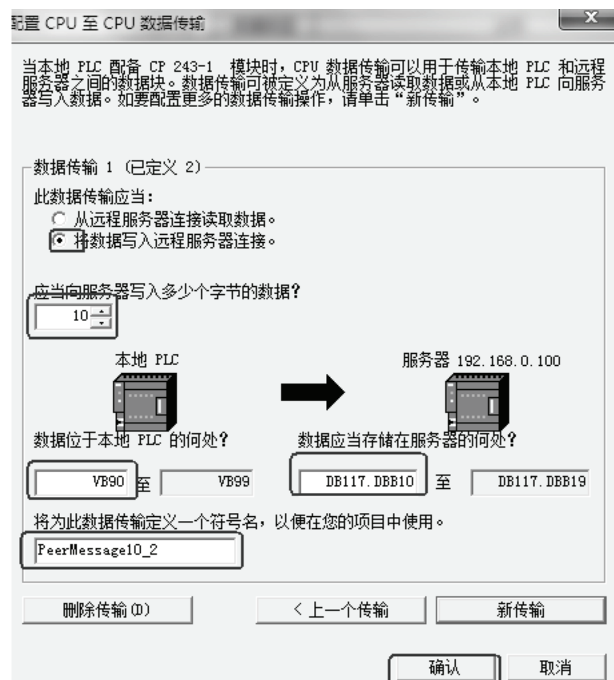


图 11-22　S7-200 PLC 发送数据

10）如图 11-23 所示，由于 CP243-1 模块的组态可能不允许再次更改，因此建议选择 CRC 保护设置，“保持活动时间间隔”保持默认设置，之后单击“下一步”按钮，弹出如图 11-24 所示界面。

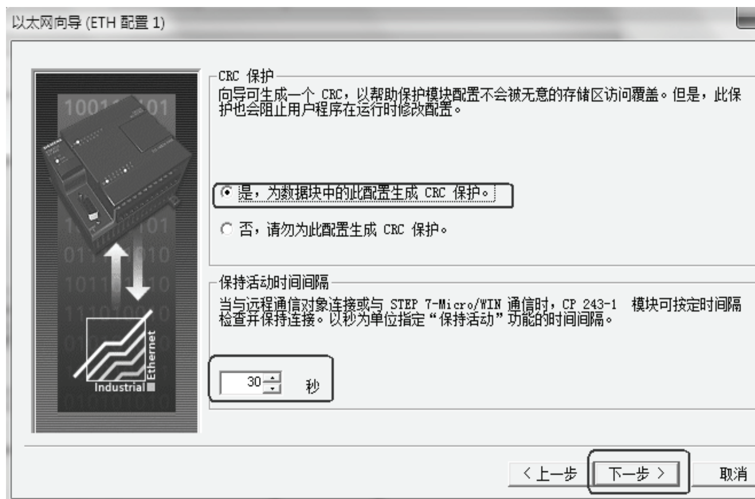


图 11-23　生成保护模块配置界面

11）在图 11-24 中，选择一个自由区域用于存储组态设置，单击“下一步”按钮完成整个以太网组态配置，如图 11-25 所示。

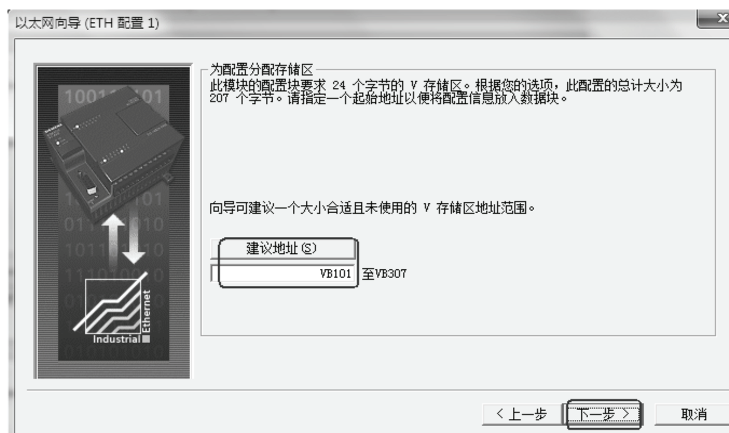


图 11-24　为以太网配置分配存储区

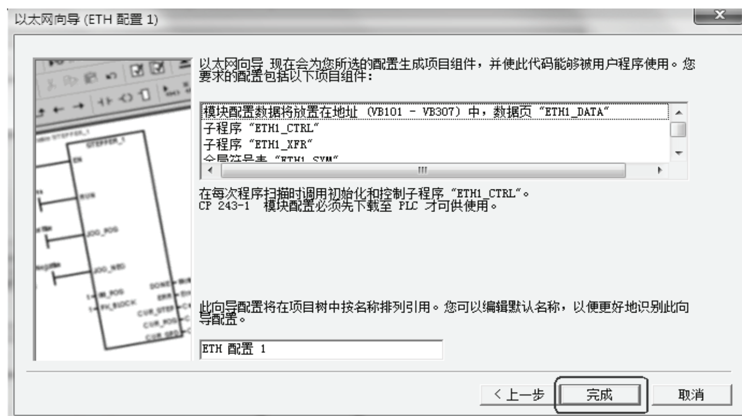


图 11-25　完成以太网模块配置任务

11.2.3 系统功能的实现

以太网向导配置完成后，将会在指令树中的“调用子程序”文件夹中创建功能块 ETHx_CTRL 和 ETHx_XTR，如图 11-26 所示。这些功能块必须在 STEP 7-Micro/WIN 软件的主程序中调用，如图 11-27 所示。

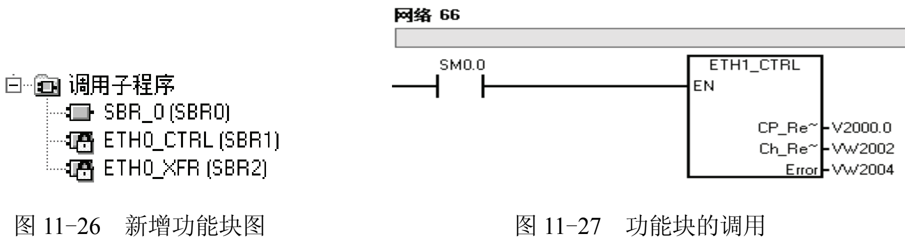


图 11-26　新增功能块图　　　　图 11-27　功能块的调用

其中，功能块 ETHx_CTRL 用于建立通信，参数性能如图 11-28 所示；功能块 ETHx_XFR 用于读写数据，参数性能如图 11-29 所示。

	符号	变量类型	数据类型	注释
	EN	IN	BOOL	
		IN		
		IN_OUT		
L0.0	CP_Ready	OUT	BOOL	CP243-1 模块准备就绪
LW1	Ch_Ready	OUT	WORD	通道准备就绪位
LW3	Error	OUT	WORD	错误字
		OUT		
		TEMP		

图 11-28　ETHx_CTRL 模块参数含义

	符号	变量类型	数据类型	注释
	EN	IN	BOOL	
L0.0	START	IN	BOOL	如果 CP243-1 不忙，向其发送命令
LB1	Chan_ID	IN	BYTE	客户机通道 ID 号码 (0-7)
LB2	Data	IN	BYTE	需要发送的信息 ID 号码 (0-31)
L3.0	Abort	IN	BOOL	1 = 取消
		IN		
		IN_OUT		
L3.1	Done	OUT	BOOL	当 CP243-1 完成命令时为‘1’
LB4	Error	OUT	BYTE	来自 CP243-1 模块的错误状态
		OUT		
		TEMP		

图 11-29　ETHx_XFR 模块参数含义

图 11-30 所示为 S7-200 PLC 程序，ETHx_XFR 连接调用两次服务器，分别用于读取 S7-300 PLC 数据和写入数据到 S7-300 中；数据交换按照配置界面设定的通道和对应地址进行。

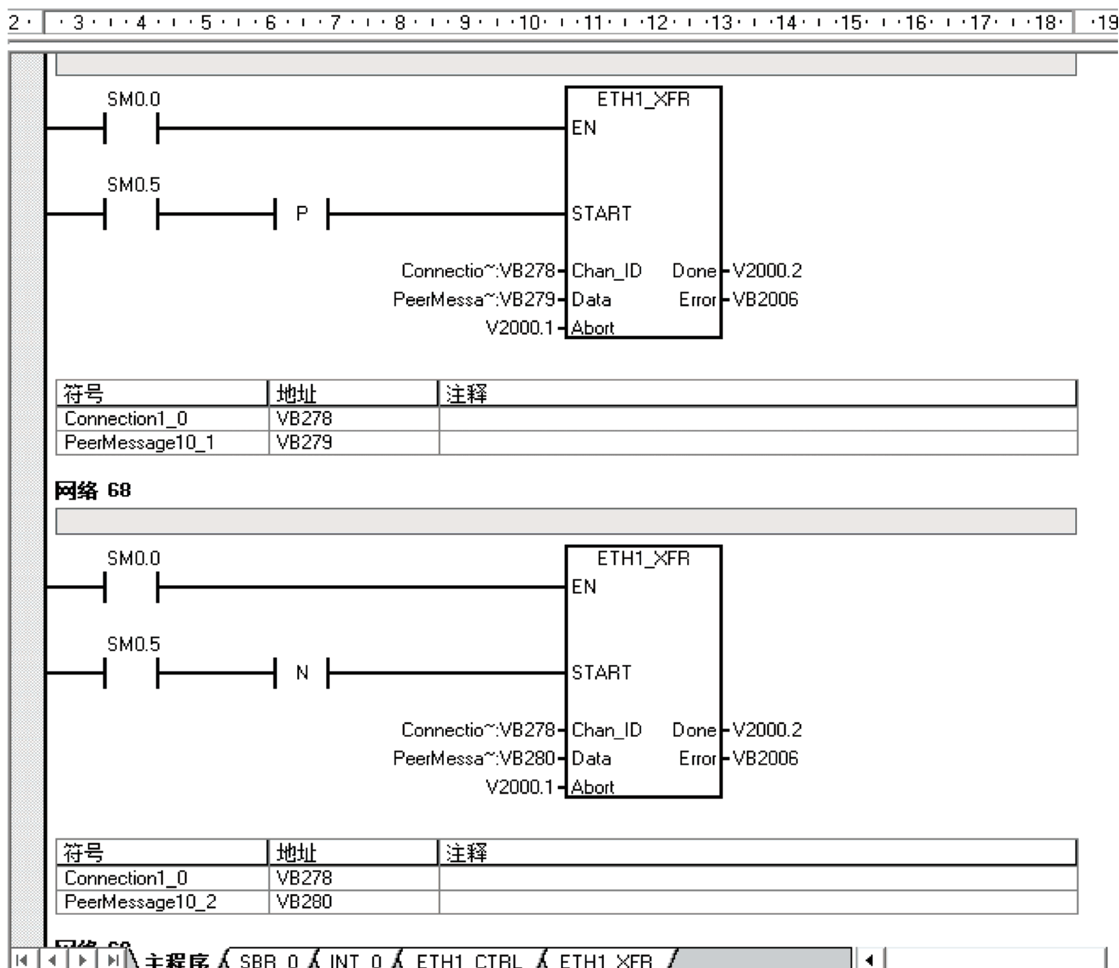


图 11-30　S7-200 PLC 与 S7-300 PLC 的通信内容

第 12 章　基于 WinCC 的异构网络系统集成

12.1　WinCC 软件介绍

WinCC 是由西门子公司开发的上位机组态软件，主要用于对生产过程进行监控。WinCC 软件基于 Microsoft 公司的 Windows 2000 或 Windows NT 操作系统。

在 WinCC 中，通信伙伴可以是自动化系统中的中央模块和通信模块，也可以是 PC 中的通信处理器，通过传送数据主要满足以下不同的用途：

1）控制生产过程。

2）调用过程数据。

3）指示过程中的异常状态。

4）归档过程数据。

WinCC 管理器界面如图 12-1 所示。

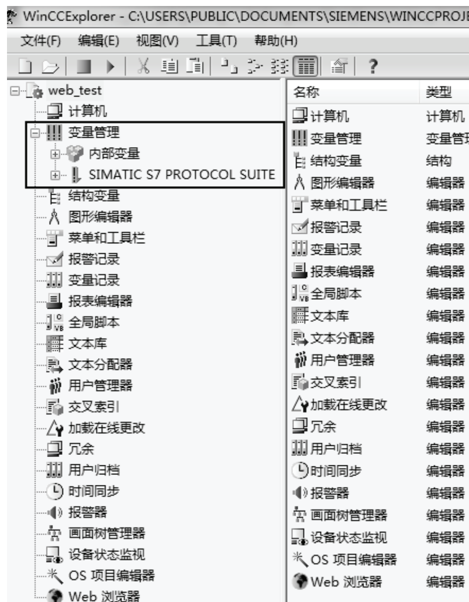


图 12-1　WinCC 界面

WinCC 组态的基本准则如下：

1）采集周期和更新时间：组态软件中定义的变量采集周期是决定可实现的更新时间的主要因素；更新时间是采集周期、传输时间和处理时间之和。

2）画面：画面的刷新频率取决于要显示的数据类型和数据量，要缩短更新时间，务必为需要快速更新的对象组态更短的采集时间。

3）曲线：使用位触发的曲线时，如果在“曲线传送区域”中设置了组位，则在 WinCC

站中会更新在该区域中设置了组位的所有曲线，这些位会在下一周期复位。

WinCC 变量分为过程变量和内部变量。过程变量（也称外部变量）用于 WinCC 和自动化系统之间的通信，过程变量的属性取决于所使用的通信驱动程序，因此在变量管理器中所创建的过程变量，将具有特定的通信驱动程序、通道单元和连接；内部变量不能连接到过程，使用内部变量可对项目内的数据进行管理或将数据传送给归档。

12.2 WinCC 通信处理

如图 12-1 所示，WinCC 使用“变量管理”功能集中管理项目变量。WinCC 在运行期间会采集和管理在项目中创建的以及在项目数据库中存储的所有数据和变量。图形运行系统、报警记录运行系统或变量记录运行系统等所有应用程序必须请求来自变量管理的 WinCC 变量数据。

为了采集过程值，WinCC 通信驱动程序向自动化站发送请求报文，而自动化系统则在相应的响应报文中将所请求的过程值发送回 WinCC。WinCC 和自动化系统之间的通信如图 12-2 所示。

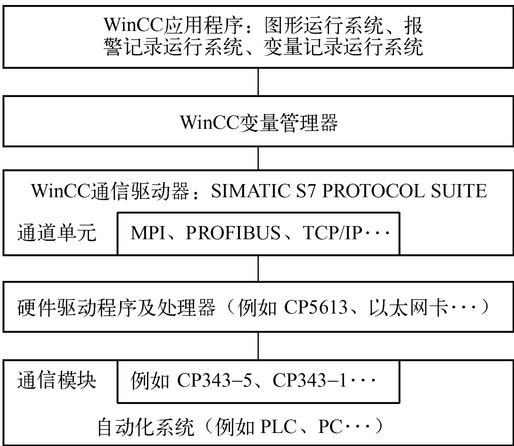


图 12-2 WinCC 和自动化系统之间的通信

WinCC 提供了许多用于通过不同总线系统连接各个自动化系统的通信驱动程序；通信驱动程序，也称为通道，其文件扩展名为“.chn”，计算机中安装的所有通道驱动程序位于 WinCC 安装目录的子目录“\bin”中，用于在自动化系统和 WinCC 的变量管理之间建立连接的软件组件，以便提供 WinCC 变量和过程值；每个通信驱动程序一次只能绑定到一个 WinCC 项目，相当于与一个基础硬件驱动程序的接口，进而也相当于与 PC 中的一个通信处理器的接口，因此，每个使用的通道单元必须分配到各自的通信处理器，针对不同通信网络会有不同的通道单元。

对 WinCC 和自动化系统进行了正确的物理连接后，WinCC 中需要通信驱动程序和相应的通道单元来创建和组态与自动化系统的逻辑连接，运行期间将通过此连接进行数据交换。

12.3 S7-200 PLC 与 S7-300 PLC 的系统集成

12.3.1 控制要求及硬件配置

系统控制要求如下：

1）S7-200 PLC 控制第一台电动机运行，S7-300 PLC 控制第二台电动机运行。

2）第一台电动机启动 5s 后第二台电动机起动。

3）如果第一台电动机起动，则第二台电动机不能单独起动、停止。

4）按下停止按钮，电动机停止。

根据控制要求，PLC 型号及系统结构可设计成如图 12-3 所示，采用以太网通信方式连接。

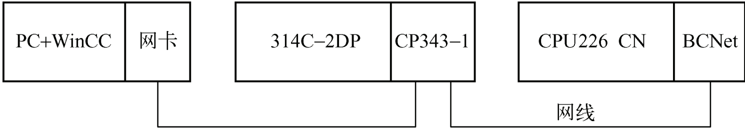


图 12-3　网络结构

其中，BCNet-S7 模块将西门子 PLC（S7-200、S7-300、S7-400）通过以太网方式连接起来，实现多站点、大规模的设备联网通信，实现设备监控和生产管理，支持 WinCC 直接连接 S7-200 PLC，无须采用 OPC 技术。

12.3.2　S7-200 PLC 与 WinCC 通信的建立

新建项目 WINCC_200_300，如图 12-4 所示，单击“变量管理”→“添加新的驱动程序”，弹出如图 12-5 所示界面，选中“SIMATIC S7 Protocol Suite.chn”，回到变量管理界面，如图 12-6 所示，选择“TCP/IP”协议，单击鼠标右键，选择“新驱动程序连接”，弹出“连接属性”界面②，单击“属性”，弹出“连接参数-TCP/IP”界面③，输入 S7-200PLC 的通信模块 BcNet-S7 PPI IP 地址，单击“确定”按钮。新的连接建立成功，名称为“S7-200”，为新连接建立变量如图 12-7 所示。

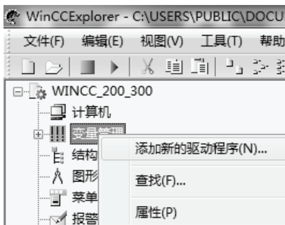


图 12-4　添加驱动

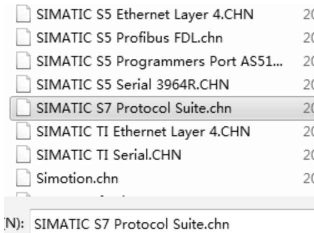


图 12-5　选择驱动类型

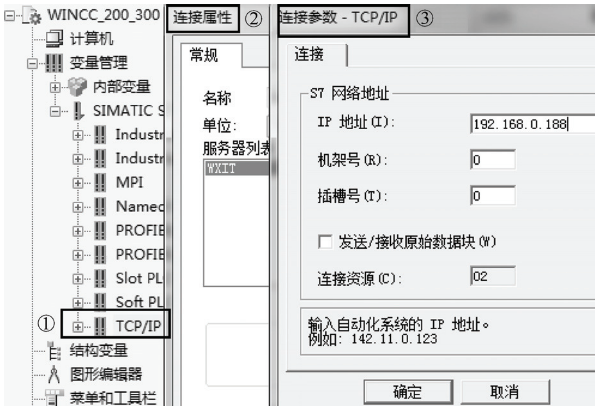


图 12-6　建立一个驱动连接

在图 12-7a 中，选中“S7-200”，在其右侧空白处右键单击“新建变量”，弹出如图 12-7b 所示界面①，选择“数据类型”，再单击“地址”栏的“选择”按钮，弹出界面②，选择数据存储器“位存储器”和存储器地址“M0.0”，单击“确定”按钮，新的变量建立完毕。

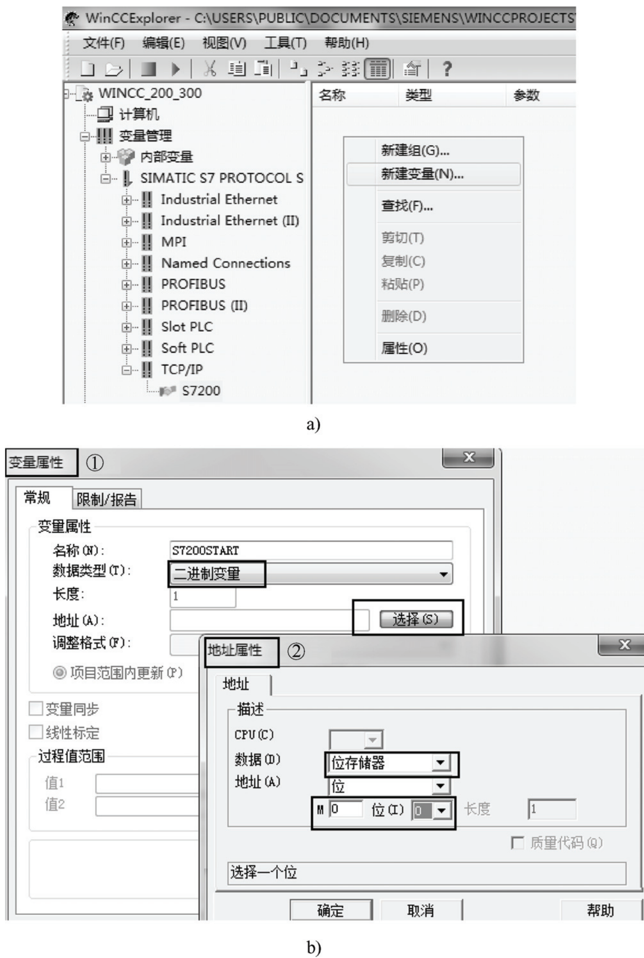


图 12-7　新建变量

S7-200 侧新建变量如图 12-8 所示，其中 M0.0 为第一台电动机起动信号，M0.1 为第一台电动机停止信号 M0.1，A0.0（即 Q0.0）为第一台电动机控制信号，也是第一台电动机运行的标志位信号。

名称	类型	参数	上次更改
S7200MOTOR	二进制变量	A0.0	2014/12/26
S7200START	二进制变量	M0.0	2014/12/26
S7200STOP	二进制变量	M0.1	2014/12/26

图 12-8　S7-200 PLC 通信变量

12.3.3 S7-300 PLC 与 WinCC 通信的建立

建立方法同上。选择“TCP/IP”协议，单击鼠标右键，选择“新驱动程序连接”，弹出图 12-9 所示“连接属性”界面②，单击“属性”按钮，弹出“连接参数-TCP/IP”界面③，输入 S7-300 PLC 的 IP 地址及 CPU 的机架号和插槽号，单击“确定”按钮。新的连接建立成功，修改新连接名称为“S7-300”，为新连接建立变量如图 12-10 所示。

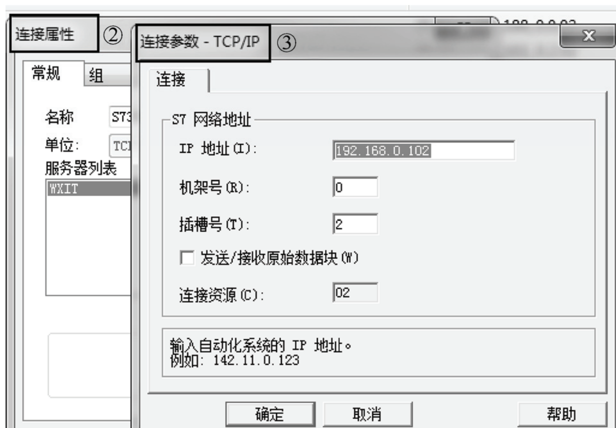


图 12-9 S7-300 PLC 通道的建立

图 12-10 中，M100.0 为第二台电动机起动信号，M100.1 为第二台电动机停止信号，M100.3 为第一台电动机运行状态信号，A0.0（即 Q0.0）为第二台电动机控制信号。



图 12-10 S7-300 PLC 通信变量

12.3.4 系统功能的实现

1．S7-200 PLC 程序的编写

根据控制要求，S7-200 PLC 程序编写如图 12-11 所示。

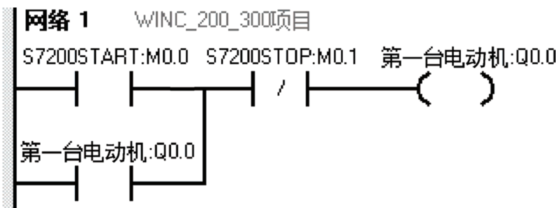


图 12-11 S7-200 PLC 梯形图

2．S7-300 PLC 程序的编写

根据控制要求，S7-300 PLC 程序编写如图 12-12 所示。

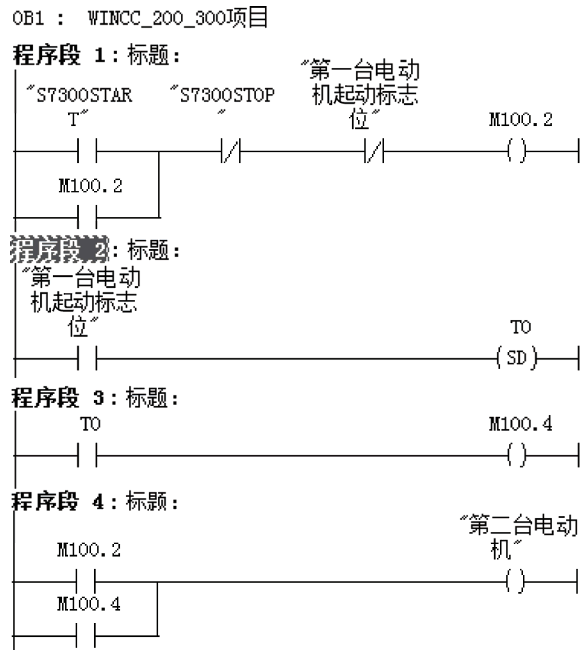


图 12-12　S7-300 PLC OB1 程序

3．全局变量的设置

根据控制要求，第二台电动机要想起动，必须判断第一台电动机的运行状态。如果第一台电动机起动，则第二台电动机延时 5s 自起动；如果第一台电动机没有运行，则第二台电动机可以在本地控制起动、停止状态。因此，S7-300 PLC 需要接收 S7-200 PLC 传递过来的第一台电动机运行状态，通信信息由 WinCC 软件实现，具体操作如下：

1）单击 WinCC 管理器“计算机”属性，按照图 12-13 所示步骤，选择“全局脚本运行系统”。

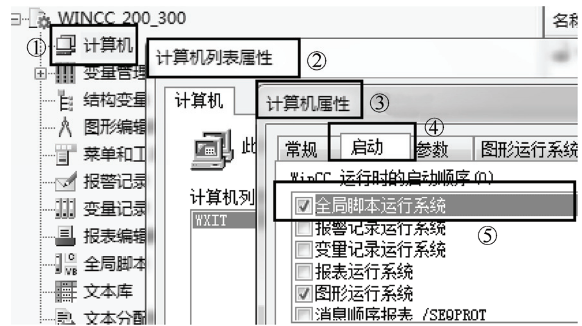


图 12-13　设置计算机属性

2）在导航窗口中，选中“全局脚本”，如图 12-14 所示。



图 12-14　全局变量路径

3）脚本编写。变量传递程序的编写如图 12-15 所示。将第一台电动机运行状态 S7-200 MOTOR 变量传递给 S7-300 PLC 的状态标志位 S7-200 LABEL，即 S7-300 PLC 程序中的“第一台电动机起动标志位”变量，由该变量起动 5s 定时器延时，从而达到第一台电动机状态控制第二台电动机起停的目的。

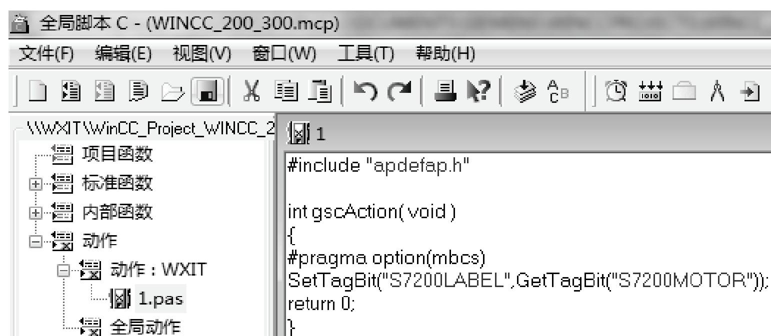


图 12-15　全局脚本程序

在采用全局变量编写程序时，还需要设置变量触发的属性，本系统设置触发动作步骤如图 12-16 所示。



图 12-16　全局变量触发动作设置

12.3.5　系统联调

分别下载 S7-200 PLC、S7-300 PLC 程序，并编制 WinCC 操作界面。程序联调及运行界面如图 12-17 所示。

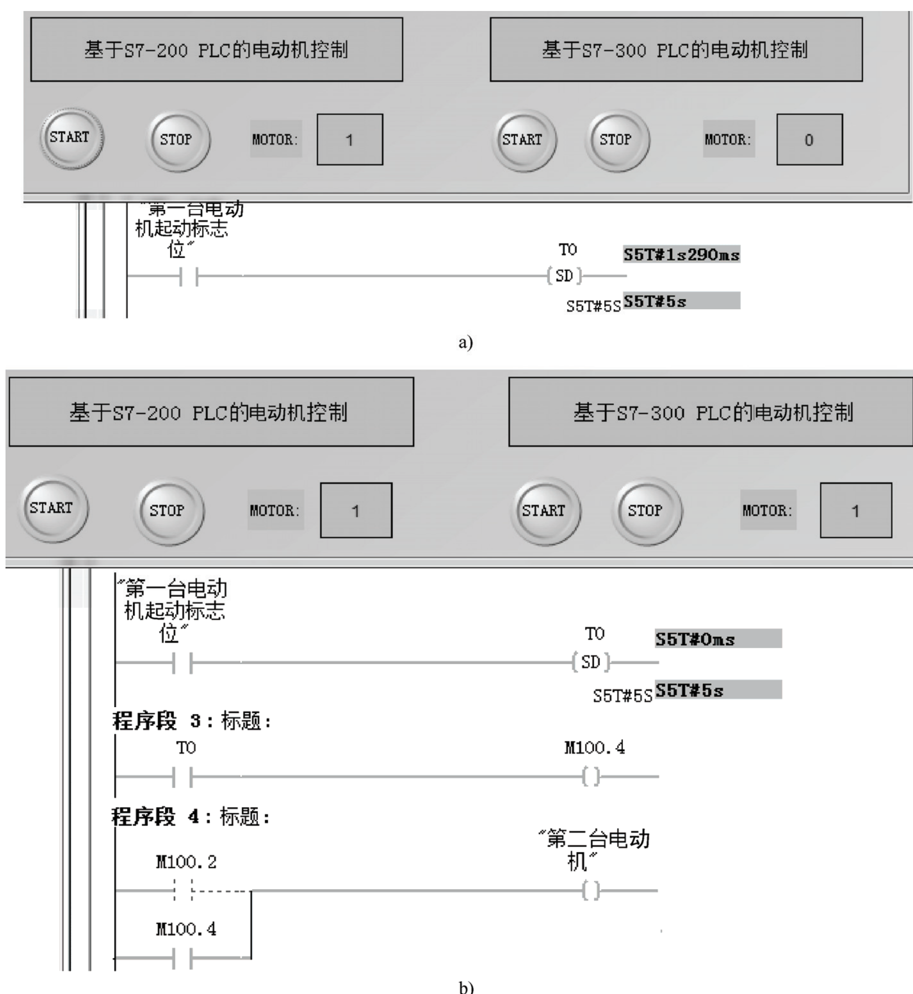


图12-17　程序联调及运行界面

a) 延时5s中　b) 5s后第二台电动机运行

12.4　三菱 Q 系列 PLC 与 S7-300 PLC 的系统集成

12.4.1　控制要求及硬件配置

WinCC 从 V7.0 SP2 版本开始增加了三菱以太网驱动程序，支持与三菱 FX_{3U}、Q 系列 PLC 进行以太网通信。

控制要求：Q PLC 与 WinCC 软件基于以太网通信，且能与 S7-300 PLC 交换数据。

硬件配置及网络结构如图 12-18 所示。

1）三菱侧。主基板：Q35B；电源模块：Q61P；CPU 模块：Q06UDHCPU；以太网模块：QJ71E71-100；以太网线。

2）西门子侧。电源模块：PS307 2A；CPU 模块：CPU315-2PN/DP；以太网线。

3）计算机侧。需要安装 WinCC 软件、STEP 7 编程软件及 GX Developer 编程软件，且需

配置网卡。

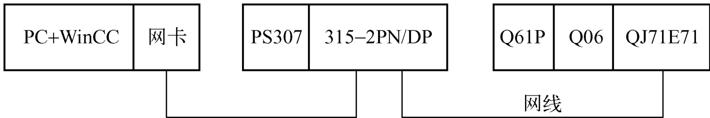


图 12–18　系统结构

同 S7-300 PLC 一样，三菱 Q 系列 PLC 也是面向中大规模复杂控制系统应用设计的，在大幅提高 CPU 模块处理性能和程序寄存器容量的同时，还提高了与网络模块、编程用外围设备之间数据通信的性能，并且可连接多达 7 块扩展基板，容纳 64 个模块；可扩展以太网模块、串口通信模块、CCLink 及其他总线模块。

12.4.2　GX Developer 软件介绍

GX Developer 编程软件，适用于三菱 Q、QnA、A、FX 等全系列可编程控制器。该编程软件支持梯形图（LAD）、指令表(STL)、顺序功能图（SFC）、功能块图（FBD）和结构化文本（ST）等多种语言程序编写，可进行程序的线上修改、监控及调试，具有异地读写 PLC 程序功能。该编程软件简单易学、易于掌握，具有丰富的工具箱和可视化界面，既可联机操作也可脱机编程，可以让设计者进行 PLC 程序的初步开发工作。

1．GX Developer 编程软件安装

1）安装通用环境。进入三菱 GX Developer 编程软件安装文件夹，找到 GX Developer/EnvMEL 文件夹，进入后单击“SETUP.EXE”，安装通用环境。

2）GX Developer 编程软件安装。返回 GX Developer 编程软件安装目录，在根目录下单击“SETUP.EXE”，根据安装向导的指引，输入相关信息和序列号即可完成编程软件的安装。

2．工程建立与程序编辑

软件安装完成后，单击运行 GX Developer 编程软件，其启动界面如图 12–19 所示。

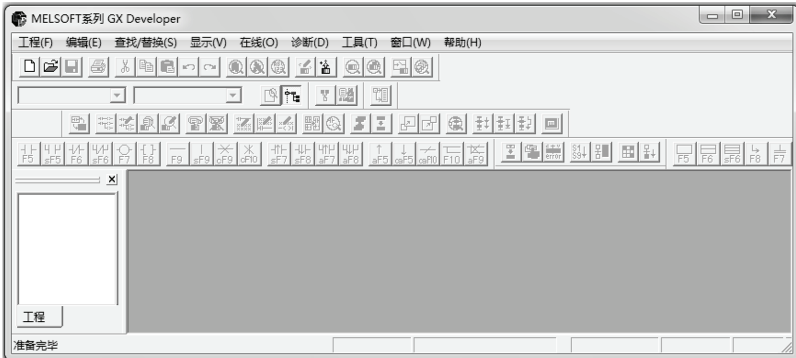


图 12–19　GX Developer 编程软件的启动界面

1）创建新工程：选择菜单栏中“工程”→“创建新工程”命令，或直接单击工具栏中的“新建”图标，可以创建一个新工程；随后按照以下步骤操作：选择 PLC 系列、类型、程序类型，设置文件名称和保存路径后，即可进入编程界面，如图 12–20 所示。注意选择的 PLC 系列和类型必须与实际使用的 PLC 一致，否则程序可能无法下载；程序类型选择梯形图；文

件名称和保存路径可自行设置。



图 12-20 采用 GX Developer 编程软件创建新工程

a) 建立新工程 b) 选择 PLC 系列 c) 选择 PLC 类型

2）编程界面：设置完成后，单击“确认”按钮，出现 GX Developer 编程软件梯形图编辑界面。编辑界面主要由菜单栏、工具栏、工程数据列表、编程区构成，如图 12-21 所示。

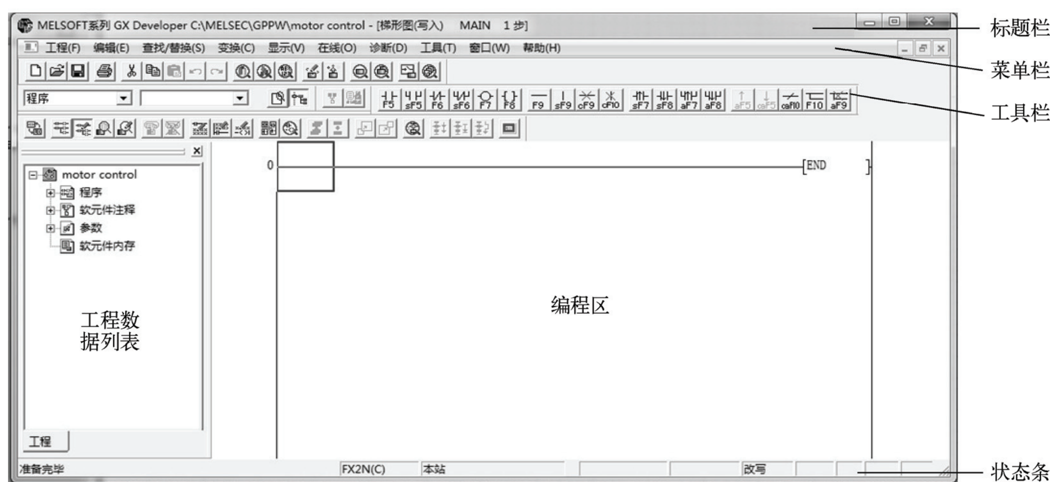


图 12-21 GX Developer 编程软件梯形图编辑界面

3．梯形图编辑

（1）创建梯形图程序

要创建梯形图程序，首先应将编辑模式设定为写入模式；当梯形图内的光标为蓝边空心框时为写入模式，可以进行梯形图的编辑，当光标为蓝边实心框时为读出模式，只能进行读取、查找等操作，可以通过选择“编辑”中的“读出模式”或“写入模式”进行切换，或用快捷键操作。

梯形图程序可采用指令直接输入法或工具按钮（快捷键）输入法。

指令直接输入法即将光标放置在需编辑的位置，然后直接输入指令，则会弹出输入窗口，按此法依次输入需编辑的程序；输入方法如图 12-22 所示。

工具按钮输入法是采用工具栏按钮或对应快捷键输入程序的方法；程序编辑时，先将光

标放置在需编辑的位置，然后单击按钮选择输入项目，在弹出的输入栏中键入元件号等，完成程序编辑。常用工具按钮及对应快捷键如图 12-23 所示，输入方法如图 12-24 所示。

图 12-22　GX 指令直接输入法示意图

图 12-23 常用工具按钮及对应快捷键图

图 12-24　工具按钮输入法示意图

（2）转换已创建的梯形图程序

已创建的梯形图程序需要经过转换处理才能进行保存和下载；单击菜单栏中的“变换”或工具栏中的按钮，也可以直接按快捷键 F4 进行变换。变换后可看到编程内容由灰色转变为白色显示，如转换中有错误出现，出错区域将继续保持灰色，需检查程序并修改正确后再次转换。

（3）梯形图的修改

GX Developer 编程软件提供了多种梯形图修改工具，用户可根据需要合理使用；主要包括插入、改写功能，剪切、复制功能及划线功能等。

对梯形图的插入或改写，可使用软件的插入、改写功能，该功能显示在软件界面的右下角，可通过电脑键盘上的 Insert 键调整；利用剪切、复制功能，可删除或移动部分程序；划线和划线删除可调整程序结构和各元件的连接关系。

12.4.3　Q PLC 以太网通信

1）新建项目后，按照图 12-25 所示步骤，设置 PLC 参数。

图 12-25　Q PLC 参数设置

2）网络参数设置。按照图 12–26 所示界面及步骤，设置网络类型、站号等，单击④“操作设置”，弹出界面如图 12–27 所示，并设置以太网模块 IP 地址，第一次设置 IP 地址需要用 USB 线或 RS-232 线下载网络参数。

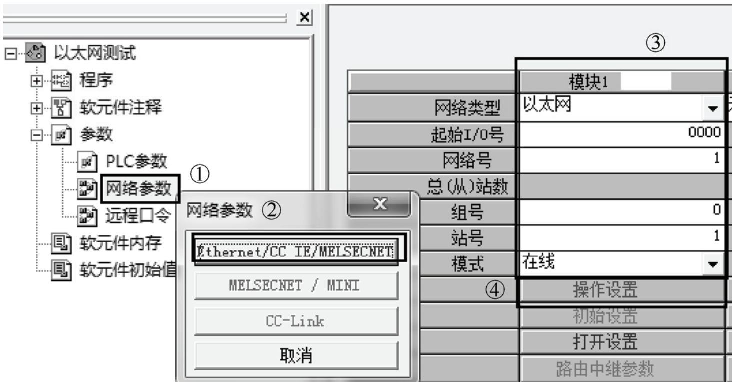


图 12–26 网络设置

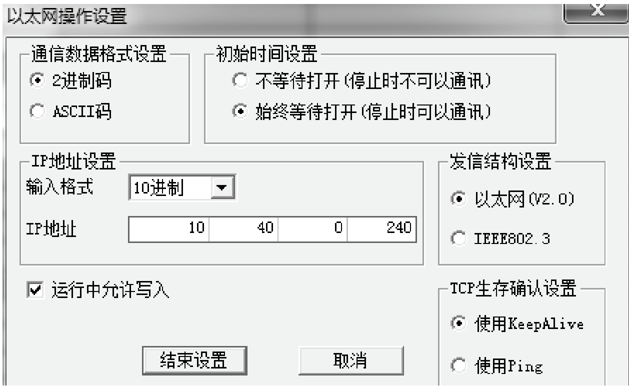


图 12–27 IP 地址设置

单击图 12–26 中的“打开设置”，弹出界面如图 12–28 所示，设置通信参数，通信协议可以选择 UDP，也可以选择 TCP，本例采用 UDP 协议。其中本站端口号为 PLC 与 WinCC 连接时的默认端口号 5001（或 5000）的十六进制数据，通信对方地址是 WinCC 安装的 PC 的 IP 地址。

	协议	打开方式	固定缓冲区	固定缓冲区通信顺序	成对打开	生存确认	本站端口号	通信对方IP地址	通信对方端口号
1	UDP		接收	有顺序	单个	不确认	1389	10. 40. 0.236	2000
2	UDP		发送	有顺序	单个	不确认	1389	10. 40. 0.236	2000
3									
4									
5									
6									

本站端口号，通信对象端口号 ： 端口号请以16进制输入。 结束设置 取消

图 12–28 通信设置

设置完毕后，将参数下载至 PLC，如图 12–29 所示。



图 12-29　网络参数下载

12.4.4　PLC 与 WinCC 通信的建立

1）按照图 12-30 所示步骤，在 WinCC 项目管理器中选择“变量管理”，添加“新的驱动程序”：Mitsubishi Ethernet.chn，并设置网络连接参数，其中在“连接属性”页面中，网络编号和 PC 编号可以对应 PLC 中的网络号和站号，也可以使用图中的默认值。

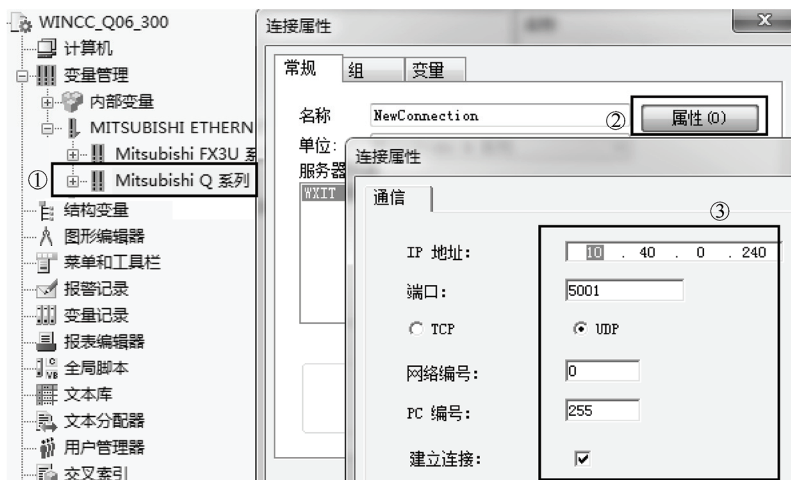


图 12-30　Q06PLC 的连接与变量建立

2）为 Q06PLC 建立通信变量 M1、M2、Y0，如图 12-31 所示。



图 12-31　Q06PLC 通信变量

3）按照图 12-30 和图 12-31 所示步骤，在 WinCC 项目管理器中选择“变量管理”，添加“新的驱动程序”：SIMATIC S7 Protocol Suite.chn，并建立通信变量 M0.1、M0.2、Q0.0，如图 12-32 所示。

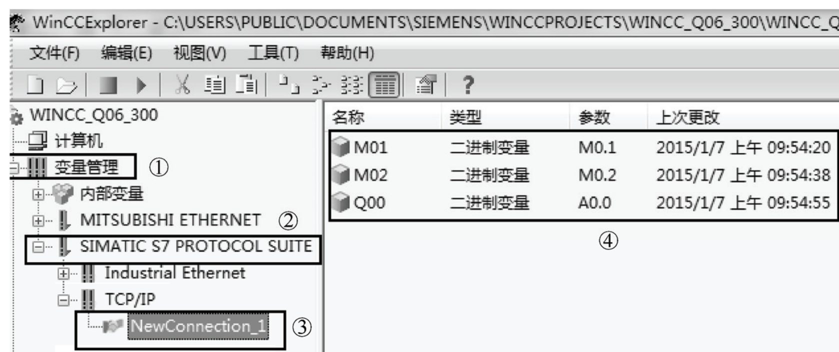


图 12-32 S7-300 PLC 的连接与变量建立

12.4.5 程序编写

1．PLC 程序

根据控制要求，Q PLC 与 S7-300 PLC 互相交换信息，因此双方会给对方发送信息，也会接收对方信息，通信信息见表 12-1。

表 12-1 PLC 之间的读写数据

Q PLC	S7 300PLC
发送数据（M1）	接收数据（M0.2）
接收数据（M2）	发送数据（M0.1）

程序编写如图 12-33 所示。

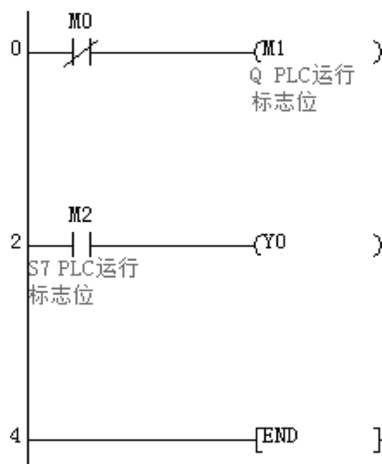
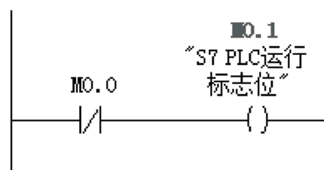


图 12-33-1 Q PLC 程序图



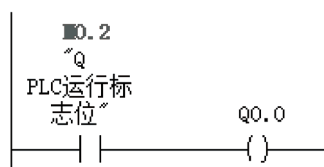


图 12-33-2 S7 PLC 程序

2．脚本程序

变量传递程序编写如图 12-34 所示。

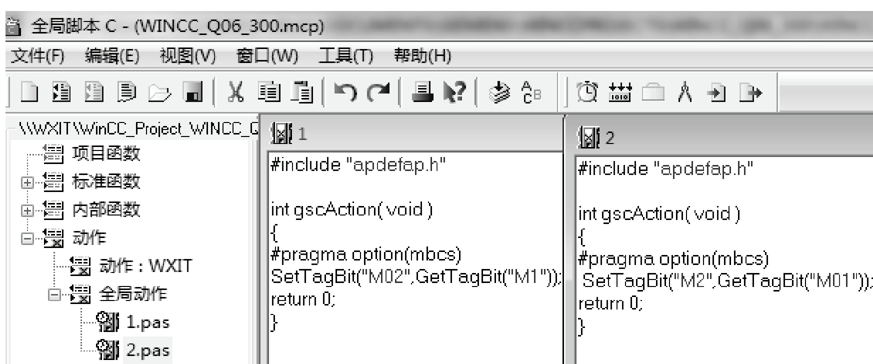


图 12-34　全局脚本程序

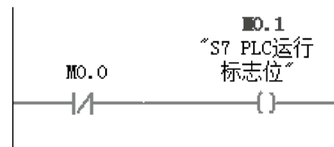
在采用全局变量编写程序时，还需要设置变量触发的属性，本系统触发动作分别设置为变量 M1（动作 1）和 M01（动作 2）。

12.4.6　系统联调

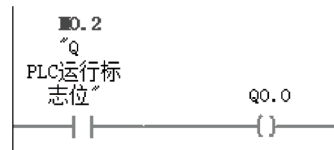
分别将程序下载至对应的 PLC 中，并运行 WinCC，观察 PLC 运行程序如图 12-35 所示，可见 M0.2 获得来自 Q PLC 的 M1 信号状态，M2 获得来自 S7 PLC 的 M0.1 信号状态，实现了两台 PLC 基于 WinCC 组态软件的通信功能。

OB1 :"Main Program Sweep (Cycle)"

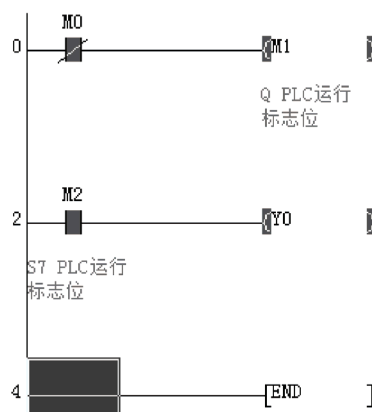
程序段 1：标题：



程序段 2：标题：



a)



b)

图12-35　PLC运行程序

a) S7 PLC程序　b) Q PLC程序

第 13 章　基于 OPC 技术的异构网络系统集成

13.1　KEPServer 软件介绍

随着 OPC 技术的不断发展和使用，各厂家纷纷针对各自的硬件推出专用的 OPC 服务器、OPC 数据访问中间件或者开发工具，极大地加速了 OPC 技术在工业控制领域的推广。

KEPServerEX 是市面上应用非常广泛的 OPC 服务器之一。它采用了业界领先的驱动程序插件式结构，在服务器中嵌入了 100 多种通信协议，不仅支持工业市场上广泛采用的数百种设备型号，还能通过下载新的驱动程序插件进行扩展，真正实现了传统 OPC 服务器所不具备的通用性。

KEPServerEX 服务器显著的特点是通过单一的服务器接口使 OPC 多协议技术得到了极大的丰富。多协议技术是指 KEPServerEX 服务器在安装过程中可以添加多种通信协议即驱动程序，而上位机客户端只需与服务器暴露的接口建立通信连接，即可获得服务器内所有数据，实现一种服务器同时组态多种不同硬件设备的功能。

KEPServerEX 支持串行、以太网连接等一系列应用最广泛的工业控制系统，包括 A-B、GE、HONEYWELL、MITSUBISHI、SIEMENS、OMRON、TOSHIBA 等厂商的各类产品。

如图 13-1 所示，KEPServerEX 服务器由对象、接口和驱动插件组成，其中对象又分为服务器对象、组对象和项对象。同时具有简单的客户端功能，能直接对硬件设备进行数据操作而不依赖客户端和硬件设备运行。应用程序遵循 OPC 技术规范对服务器进行各项操作，如通过读/写函数的调用对实际设备中的数据标签进行读写。标签项代表该标签到现场数据源的逻辑连接，在标签创建时定义。

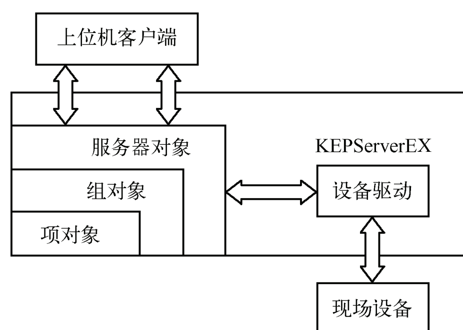


图 13-1　KEPServerEX 服务器结构

KEPServerEX 服务器对象内提供了管理多个组对象的方法，通过接口调用对其进行各种操作。服务器对象由多个组对象组成，每个组对象对应一个通道，每个通道都有自己的驱动，通道之间互不影响；设备驱动定义了从现场设备获取数据的操作方法，由于不同厂家设备的

通信协议和数据采集格式不同，所以驱动也不尽相同。组对象包含多个项对象，项对象定义了标签到现场数据源的逻辑连接，标签内容包括变量的类型、变量值、时间戳和通信状态等内容。

使用 KEPServerEX 服务器作为整个网络系统的数据服务器，可以实现对多个子系统生产数据的统一采集，同时通过客户端访问服务器中任意子系统的现场数据。

13.2 SIMATIC NET 软件介绍

SIMATIC NET 可以用于实现与西门子 PLC 的各种协议通信，是将西门子的产品通过 SIMATIC NET 集成到整个自动化领域的软件，提供的 OPC 服务器可以与任何支持 OPC 通信的产品进行数据交换，例如 WinCC、InTouch、iFix 等，用户自己编写的 OPC 客户端程序也可以。其功能如下：

1）包括几乎所有西门子软硬通信网卡的驱动程序，如 CP5613、CP1613、CP1613A2、CP5611、CP5512 等，STEP 7 集成的“设置 PG/PC 接口”工具不支持上述通信卡，将这类卡插入计算机，在安装 SIMATIC NET 时，将自动地为它们安装驱动程序。如图 13-2 所示，计算机上装有 CP5613A2 卡，当安装 SIMATIC NET 时，在 STEP 7 的“设置 PG/PC 接口”设置中就可以选择该卡通信驱动方式。

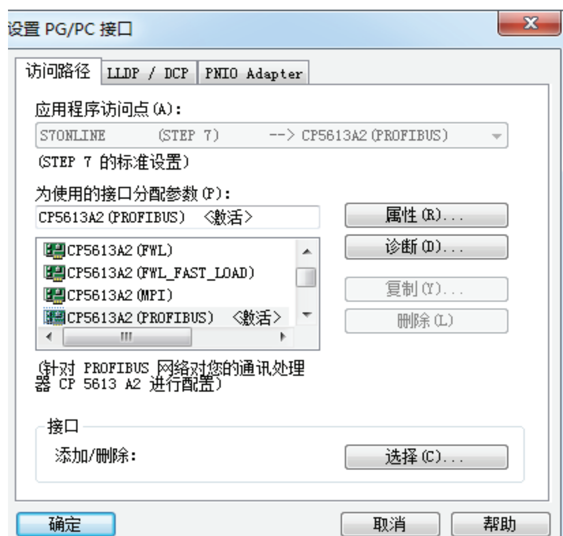


图 13-2　CP5613A2 卡通信驱动选择

2）包括 NCM（Net Configration Manager）网络组态管理软件和 SCE（Station Configuration Editor）PC 站组态软件，与 STEP 7 中的 HW Config 和 NetPRO 类似。

3）包括 SIMATIC NET OPC Server 工具软件，通过 OPC 可以和第三方设备进行通信。如果上位计算机运行的组态软件不支持西门子的通信协议，不能直接访问西门子 PLC，需要用 SIMATIC NET 的 OPC 功能来解决这一问题。

对于国内及国外的组态软件，在访问西门子的 PLC 时，有时候需要 SIMATIC NET，有

时候不需要，主要是看相应组态软件开发了哪些协议，如果开发了，就不需要通过 SIMATIC NET，可以直接访问 PLC，否则需要通过 SIMATIC NET 配置 PC Satation，然后通过 OPC Server 或者 Application 接口来访问。

SIMATIC NET 软件提供了统一的 Windows 画面，通过它提供了 APC（Advanced PC Configuration）高级 PC 配置工具，PC 可以作为整个系统控制系统的一个站点同其他 PLC 站进行通信，SIMATIC NET OPC 是服务器/客户端结构，客户端访问服务器的程序。采用 OPC 技术大大提高了控制系统的互操作性和适应性。

13.3 基于 KEPServer 的 FX_{2N} PLC 与 S7-300 PLC 的通信系统

13.3.1 控制要求及硬件配置

控制要求：S7-300 PLC 控制的电动机起动 5s 后，FX_{2N} PLC 控制的电动机起动，要求 S7-300 PLC 定时时间到后发给 FX_{2N} PLC 电动机起动信号。

分析：考虑两台 PLC 是不同种类的 PLC，且通信口协议不同，因此采用 OPC 服务器实现异构网络 PLC 之间的通信。

硬件配置：装有 KEPware OPC Server V4.5 的计算机 1 台；CPU315 2PN/DP PLC 1 台；FX_{2N} -48MR PLC 1 台；网线 1 根、USB-SC09 编程电缆一根。

13.3.2 S7-300 PLC 与服务器的通信建立

打开 KEPServerEX 软件，新建工程“test”，按照操作界面提示进行操作。例如在图 13-3 中，单击鼠标右键，弹出“New Channel”界面，按照图示提示操作，进入图 13-4 所示界面。



图 13-3 输入 Channel name

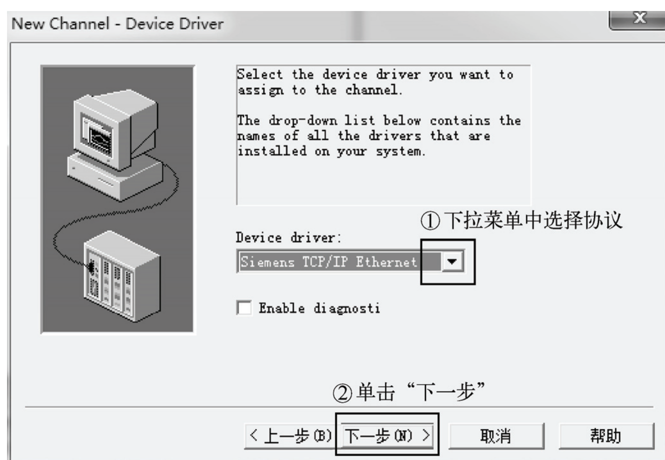


图 13-4　设备驱动选择

图 13-4 所示界面为设备驱动界面，本系统中 CPU315 PLC 的 PN 接口与计算机的 RJ45 接口通过网线连接，因此在下拉菜单中选择“Simens TCP/IP Ethernet”，单击“下一步”按钮，进入新界面，选择默认值，直到弹出图 13-5 所示界面，单击“完成”按钮，通道建立完毕。

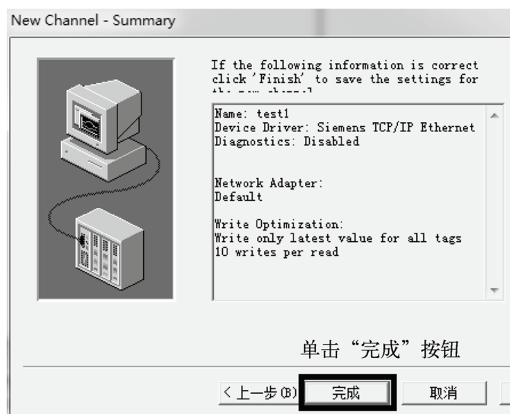


图 13-5　通道信息

在图 13-6 中添加本通道中的设备信息，单击“下一步”按钮，弹出图 13-7 所示界面，选择设备类型，单击“下一步”按钮，弹出图 13-8 所示界面，输入所选设备 IP 地址，单击“下一步”按钮，进入新界面，选择默认值，直到弹出图 13-9 所示界面，单击“完成”按钮，设备信息添加完毕。

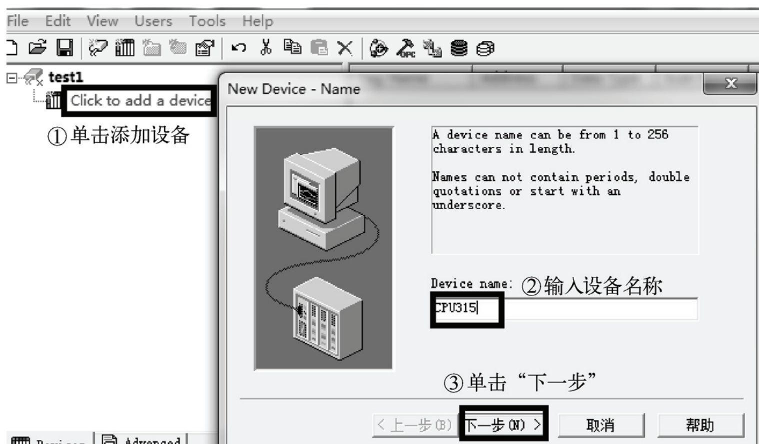


图 13-6　添加设备操作

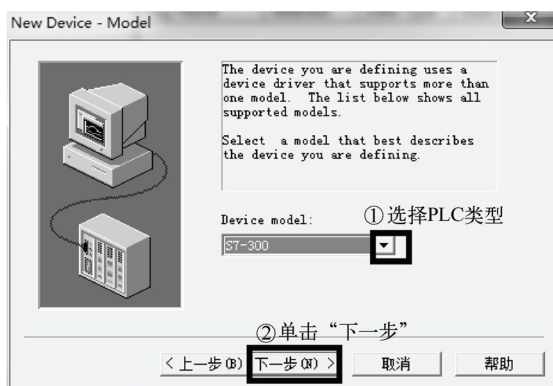


图 13-7　设备类型

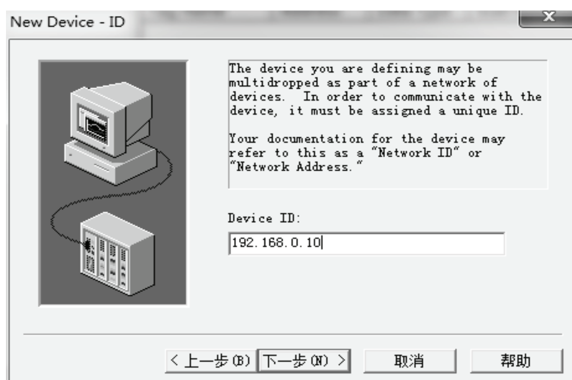


图 13-8　输入设备 IP 地址

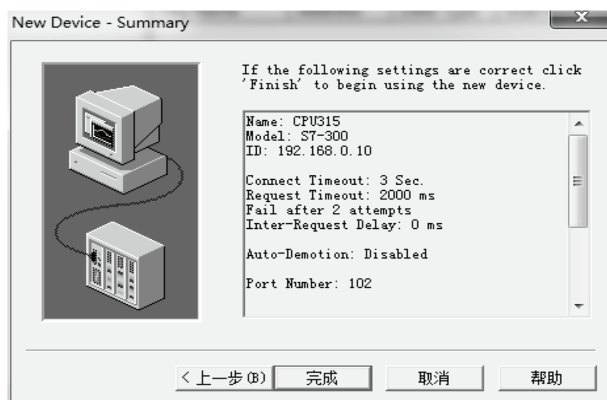


图 13-9　设备信息

13.3.3　通信变量设置

设备信息设置完成后显示图 13-10 界面，按照界面提示，在图右侧建立系统需要的变量信息，如图 13-11 所示，单击“New Tag”选项，弹出图 13-12 所示界面，填写变量属性。

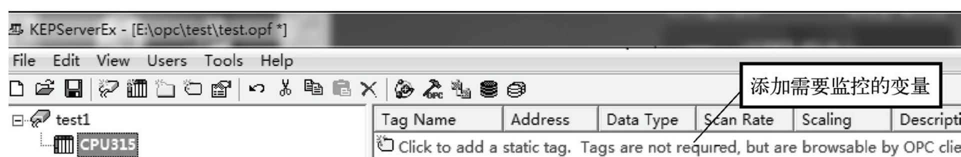


图 13-10　在 OPC 客户端添加变量

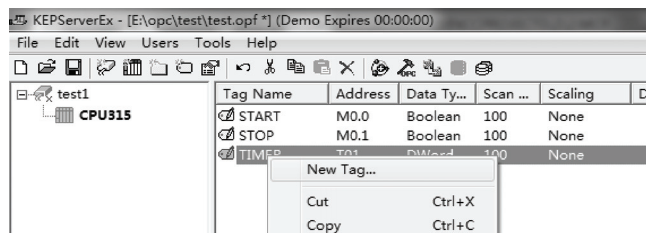


图 13-11　添加变量

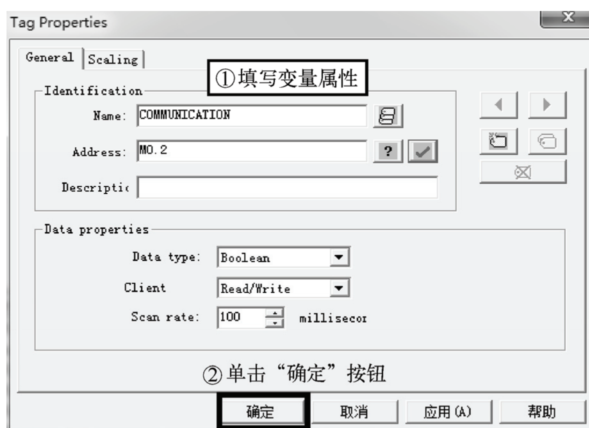


图 13-12　设备变量的建立

13.3.4　PLC 与 KEPServer 系统联调

单击快捷键 Quick Client，弹出 OPC Quick Client 页面，如图 13-13 所示，从图中可见，OPC 服务器与建立的 test1 客户端通信状态良好，系统运行正常。

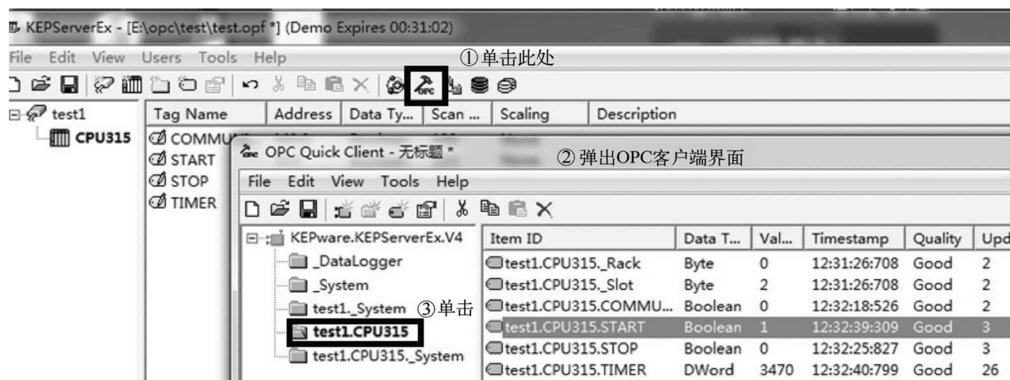


图 13-13　CPU315 PLC 客户端的通信连接与变量观察

FX_{2N}-48MR PLC 在 KEPware 软件中建立通道、设备设置和变量添加可参考 13.3.2 节和 13.3.3 节，其变量建立如图 13-14 所示，变量观察如图 13-15 所示。

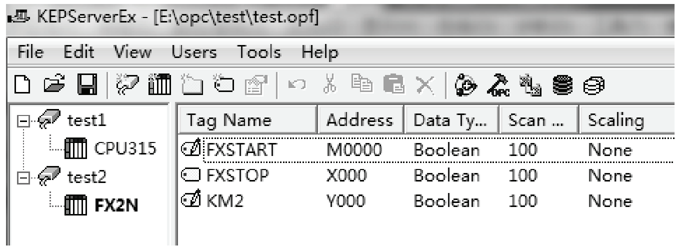


图 13-14 FX_{2N}-48MR PLC 客户端变量的建立

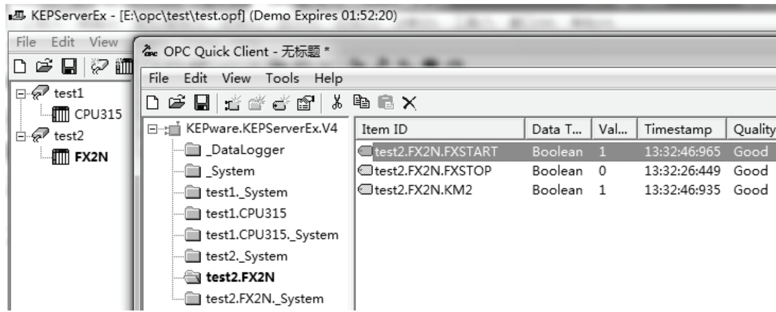


图 13-15 FX_{2N}-48MR PLC 客户端的通信连接与变量观察

13.3.5 WinCC 与 KEPServer 通信建立

WinCC 软件可以作 OPC 服务器，也可以作 OPC 客户端，当使用 WinCC 作为 OPC 客户机时，必须将 OPC 通道 OPC.CHN 添加到 WinCC 项目中，如图 13-16 所示操作，打开 WinCC Explorer 软件，进入主界面，在新建项目“testproject”的“变量管理”中添加 OPC 驱动。

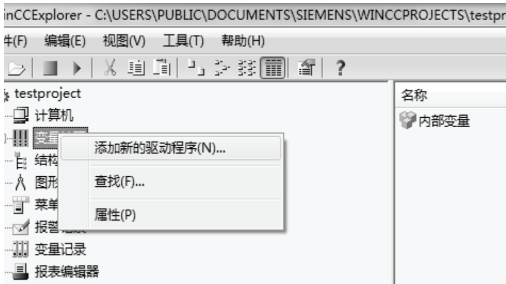


图 13-16 添加 OPC 驱动

OPC 驱动添加成功后出现图 13-17 所示界面，右键单击选择“系统参数”，弹出界面如图 13-18 所示，添加一个与该驱动程序相连接的设备。每一个 OPC 服务器都拥有自身可编址的 ProgID（程序 ID），通过 OPC 条目管理器，可请求 OPC 服务器 ProgID。展开“\\<LOCAL”，选择前面所述步骤建立的本地服务器“KEPware.KEPSeverEx.V4”，单击“浏览服务器”按钮，出现界面如图 13-19 所示，单击“下一步”按钮，进入界面如图 13-20 所示。

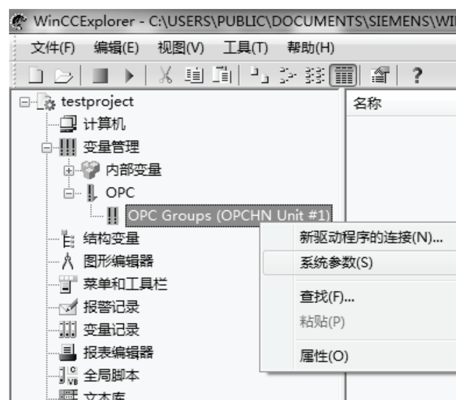


图 13-17　OPC 参数设置

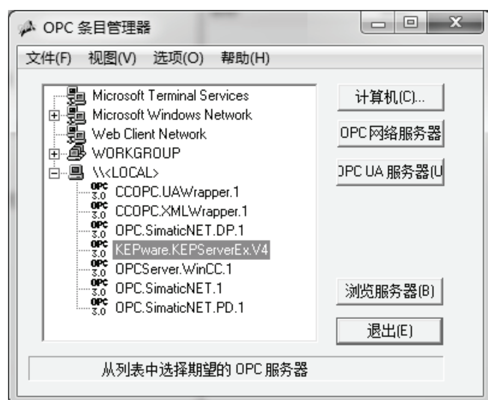


图 13-18　选择 OPC 服务器的 ProgID

在图 13-20 中，选择“test1”→CPU315，出现右侧变量，选择所有变量，单击“添加条目”按钮（同理添加“test2”所有变量）；出现界面如图 13-21 所示，选择“KEPware_KEPSeverEx_V4”，单击“完成”按钮，变量添加完成情况如图 13-22 所示界面。



图 13-19　过滤标准界面



图 13-20　进入 KEPware 服务器



图 13-21　添加变量界面



图 13-22　OPC 服务器变量

13.3.6　PLC 之间的数据交换

在图 13-22 中，选中导航窗口中的“计算机”，单击鼠标右键，进入“计算机属性”界面，选中“全局脚本运行系统”，单击“确定”按钮结束。

在导航窗口中，选中“全局脚本”，按照图 13-23 所示步骤编写动作函数，将 S7-300 PLC 中的标志位 COMMUNICATION 状态传递给 FX_{2N} PLC 的起动变量 FXSTART。观察组态界面，

如图 13-24 所示，当按下 S7 300PLC 连接的起动按钮 START 后，KM1 状态为 1，驱动第 1 台电动机，定时器开始计时；当 5s 时间到后，标志位 COMMUNICATION = 1，FX_{2N} PLC 的起动变量 FXSTART=1，则 KM2=1，驱动第 2 台电动机工作，如图 13-25 所示。

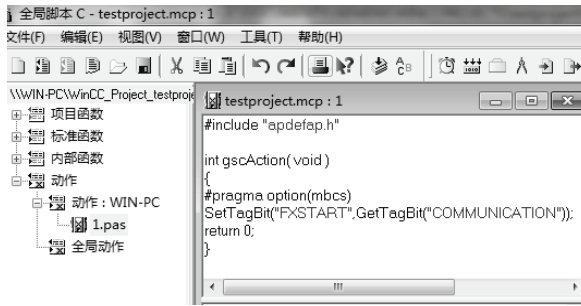


图 13-23　编写动作函数

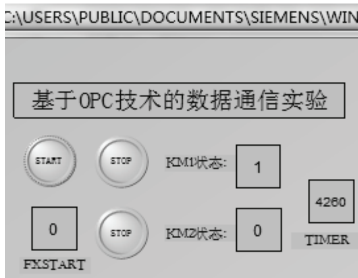


图 13-24　监控界面 1

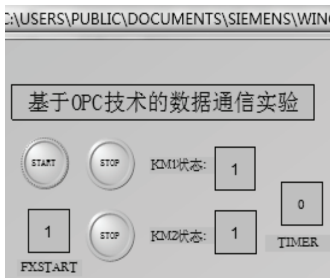


图 13-25　监控界面 2

13.4　基于 SIMATIC NET 的 OPC 服务器与 S7-300 PLC 的通信系统

13.4.1　控制要求及硬件配置

控制要求：采用 OPC 技术实现 OPC 服务器与 S7-300 PLC 之间的通信建立。在上位机操作界面，按下起动按钮，第一台电动机起动，延时若干时间后第二台电动机起动；延时时间可调，按下停止按钮两台电动机都停止。

硬件配置：PC 1 台，CPU315-2PN/DP 1 块，普通网卡 1 块，网线 1 根，系统结构如图 13-26 所示。

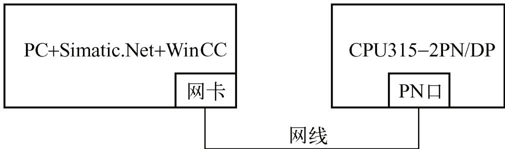


图 13-26　系统结构

13.4.2　PC 站的配置与组态

1．硬件配置

1）打开 Station Configuration Editor 配置窗口，在 1 号插槽中添加 OPC Server，按照图 13-27 所示操作步骤建立；同样方法选择 3 号槽添加 IE General，按照图 13-28 所示步骤建立。

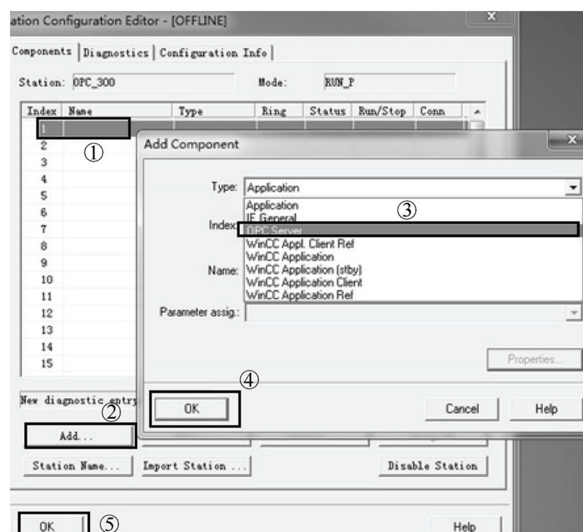


图 13-27　配置 PC 站的硬件机架（1）

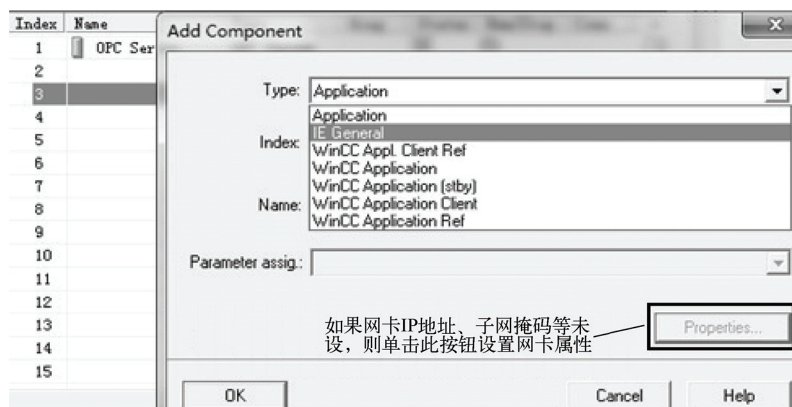


图 13-28　配置 PC 站的硬件机架（2）

在图 13-28 中，如果网卡属性已设，则“Properties”为灰色，单击“OK”按钮，弹出图 13-29 所示界面，可确认或修改网卡属性。

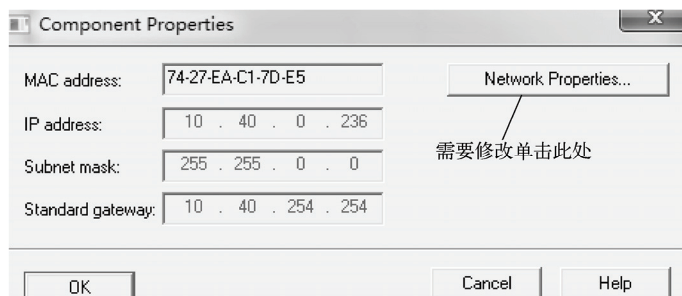


图 13-29　网卡属性

2）分配或修改 PC 站名称。该名称是项目的 PC 站名称而非计算机名称，操作步骤如图 13-30 所示。



图 13-30　分配 PC 站名称

2．PC 站的组态

1）通信设置。单击“start”→“SIMATIC”→“SIMATIC NET”→“Siemens Communication Settings”，弹出界面如图 13-31 所示，选择“Modules”→“General”，可以看见以太网模式已从“PG mode”切换到“Configured mode”，插槽号也自动指向 3。

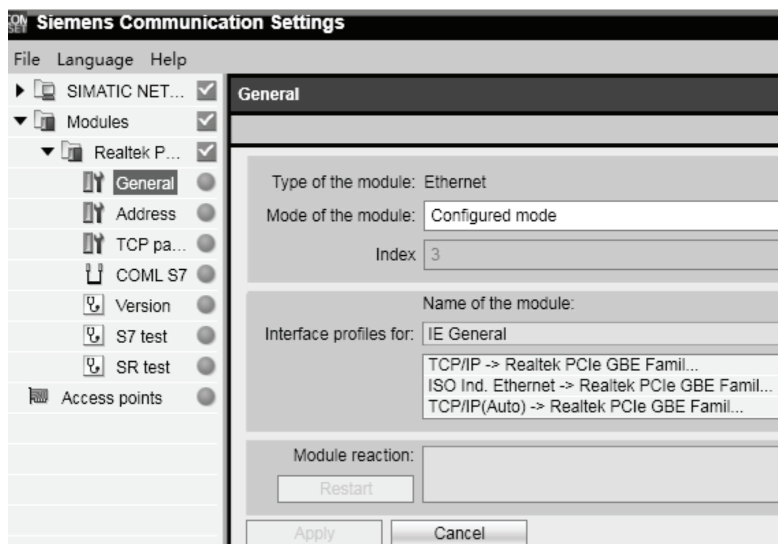


图 13-31　通信配置（1）

在 Access points 设定窗口中，将 S7ONLINE 选择指向“PC Internal”，如图 13-32 所示，单击“Apply” 按钮，设置完成，这样设定的目的是为 PC 站组态完毕下载做准备。

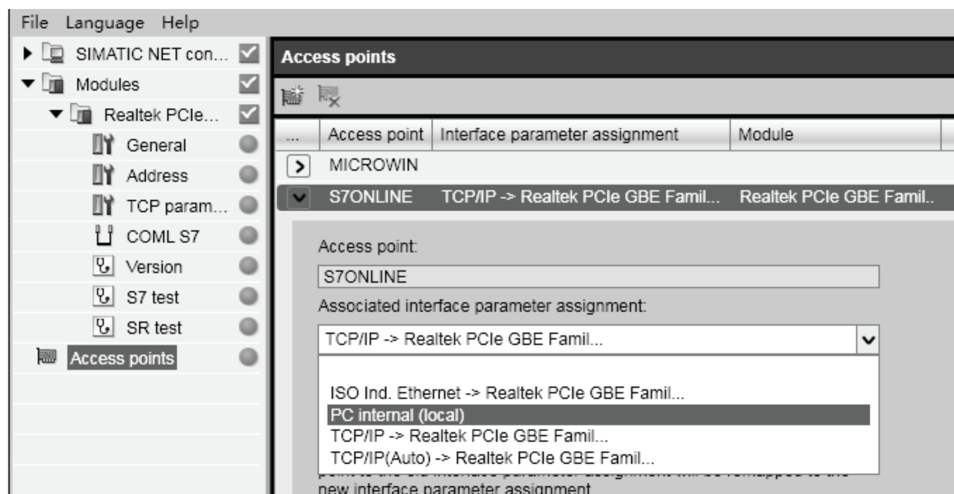


图 13-32　通信配置（2）

2）在 STEP 7 中组态 PC 站。在 SIMATIC Manager 管理界面建立项目，并插入 SIMATIC PC 站点，如图 13-33 所示，双击“组态”选项，进入 PC 站组态界面，如图 13-34 所示。

在插槽 1 中组态 OPC 服务器，操作步骤如图 13-34 所示。



图 13-33　PC 站项目的建立



图 13-34　OPC 服务器的组态

在插槽 3 中组态 IE General，组态步骤如图 13-35 所示，单击“确定”按钮后，弹出界面如图 13-36 所示。



图 13-35 IE General 的组态

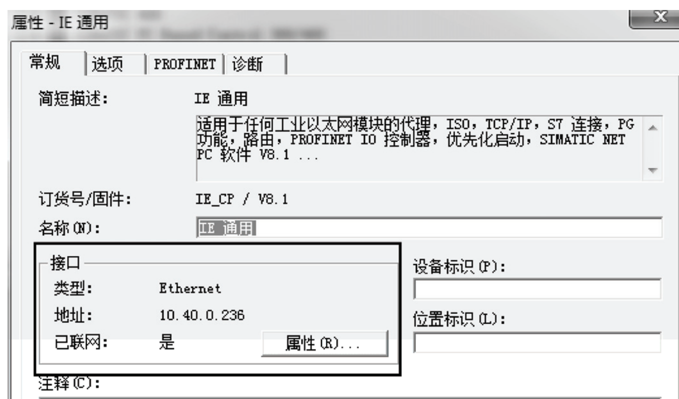


图 13-36 IE 通用属性

完成 PC 站组件设置后，单击编译存盘按钮，存储当前组态配置。

如果编译无误，单击“组态网络”按钮，如图 13-37 所示，进入“NetPro”配置窗口。



图 13-37 单击“组态网络”

在“NetPro”界面，用鼠标选择 OPC Server 后在连接表第一行单击鼠标右键弹出菜单，选择“插入新连接”，如图 13-38 所示。

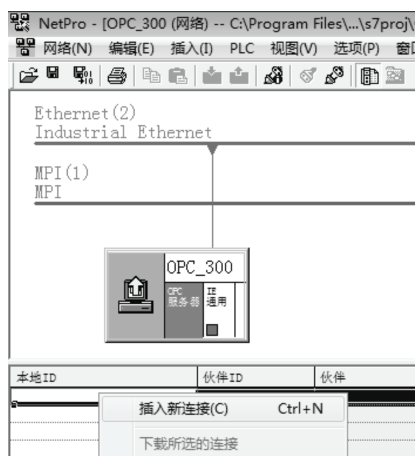


图 13-38　插入新连接

弹出如图 13-39 所示界面，按照图中标注序号步骤进行设置和操作；如果与 OPC Server 连接的 S7-300 PLC 的以太网连接已经在 STEP 7 项目中组态，则在选择“插入新连接”后，连接会自动创建，不需要再设置图中所示步骤，仅需要确认连接属性即可。

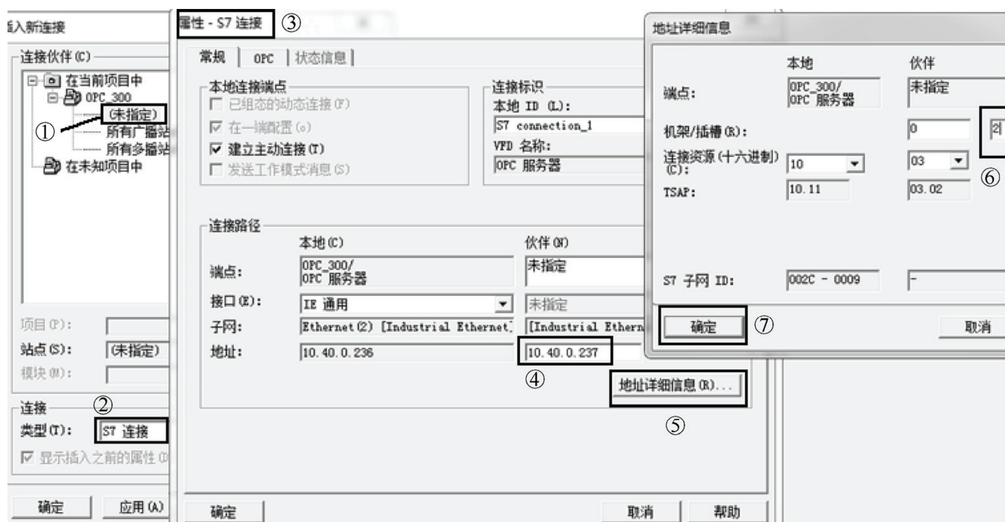


图 13-39　设置新连接参数

确认配置完成后，已建好的 S7 连接会显示在连接列表中，如图 13-40 所示。

本地ID	伙伴ID	伙伴	类型	激活的连接伙伴	子网
S7 connection_1		未知	S7 连接	是	Ethernet(2) ...

图 13-40　连接完成

单击编译存盘按钮，如果编译结果正确，则组态完成。

3．组态下载

在“NetPro”窗口单击“下载”按钮将组态下载到 PC 站中，下载模块信息如图 13-41 所

示，图 13-42 所示为确认下载信息后弹出的提示界面。

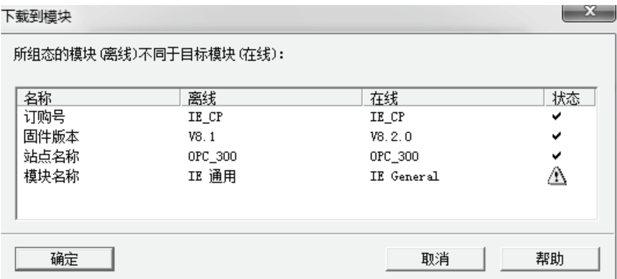


图 13-41　下载到模块的信息确认

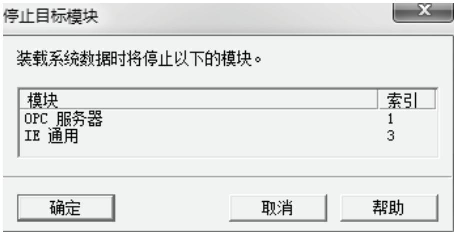


图 13-42　下载中的提示界面

下载完成后，可以打开“Station Configuration Editor”配置窗口检查组件状态，配置正确后，出现如图 13-43 所示画面，在 OPC 服务器一栏中，Conn 出现图标，说明 OPC 服务器连接激活。

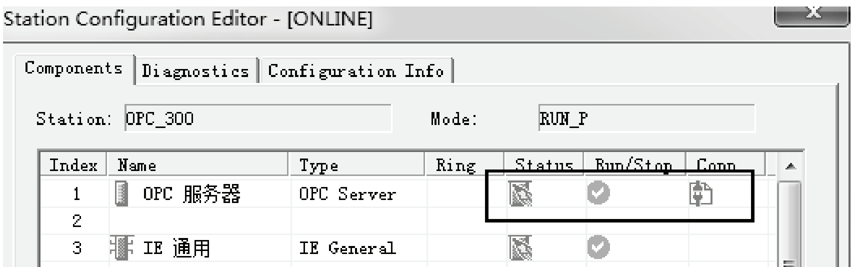


图 13-43　服务器配置成功

13.4.3　S7-300 PLC 项目设计

S7-300 PLC 硬件组态时，IP 地址应与 PC 站建立的新连接 IP 地址一致，如图 13-44 所示。在程序设计时变量分配见表 13-1。程序实现如图 13-45 所示。

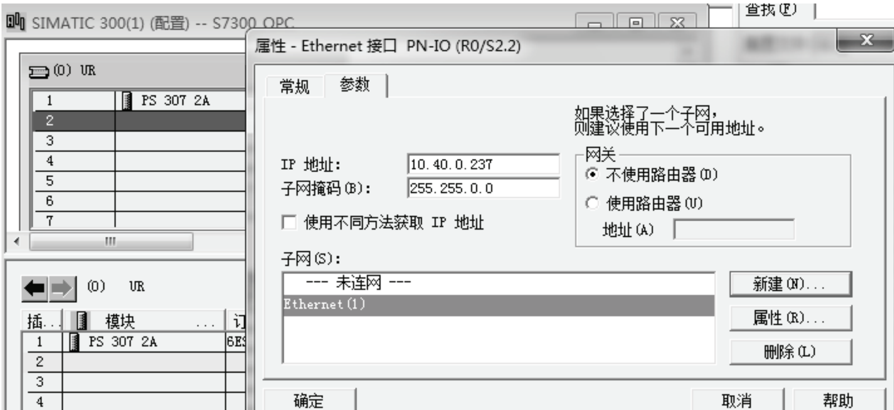


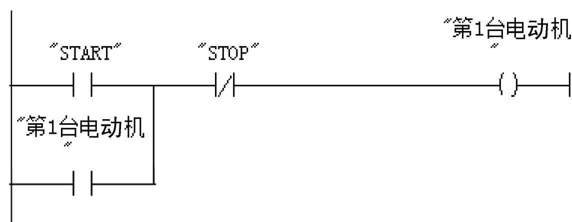
图 13-44　S7-300 PLC IP 地址设置

表 13-1　变量表

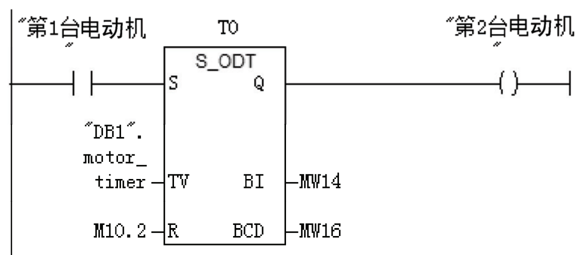
	符号	地址		数据类型
1	START	M	10.0	BOOL
2	STOP	M	10.1	BOOL
3	第1台电动机	Q	0.0	BOOL
4	第2台电动机	Q	0.1	BOOL
5	时间变量	DB	1	DB　1

OB1 : S7-300 PLC程序

程序段 1：第一台电动机起动



程序段 2：延时DB1.DBW0秒后第二台电动机起动



程序段 3：上位机界面输入MW18设定的时间，单位（s）

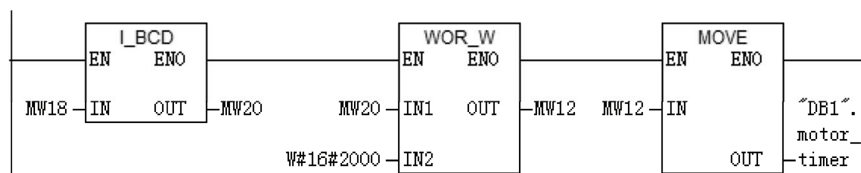


图 13-45　S7-300 PLC 程序设计

13.4.4　数据通信测试

当完成 PC 站组态下载后，可用 OPC Scout 工具进行服务器与 PLC 之间的数据通信测试，该工具随 Simatic Net 软件一起提供。建立通信变量步骤如图 13-46 所示，其中⑤⑥⑦参数必须定义。⑤是数据类型，可选择 X（位）、B（字节）、W（字）等；⑥是数据首地址；⑦是数据长度设定。同理可建立 Q 变量。

Q 变量建立完成后，出现如图 13-47 所示的界面，如果通信正常，则变量“Properties”界面如图 13-48 所示，显示通信质量状态“good”，即可以对该标签变量进行读/写操作。

在 OPC Scout 窗口中，单击“Workbook”→“Views”，建立一个观察窗口，如图 13-49 所示，观察变量状态如图 13-50 所示。

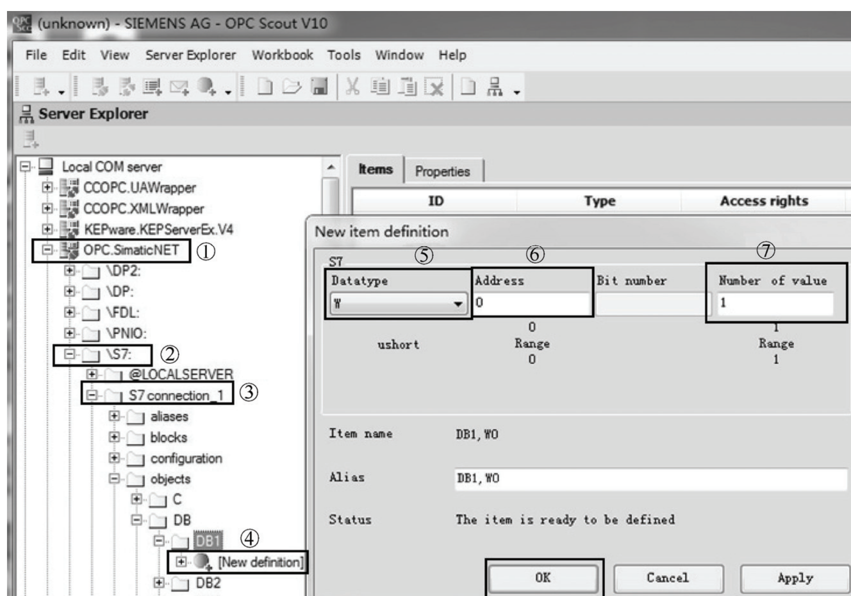


图 13-46　通信变量的建立

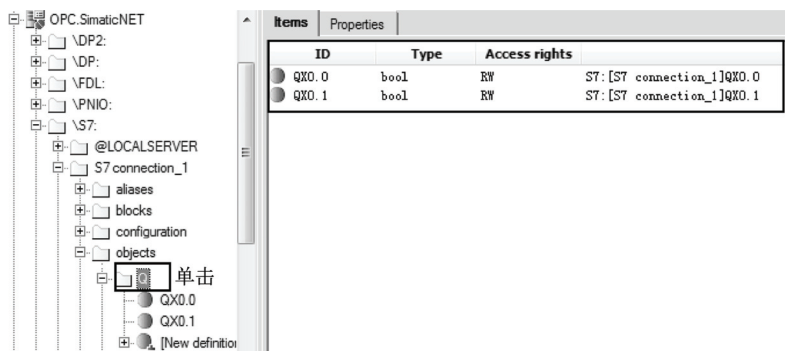


图 13-47　Q 变量列表

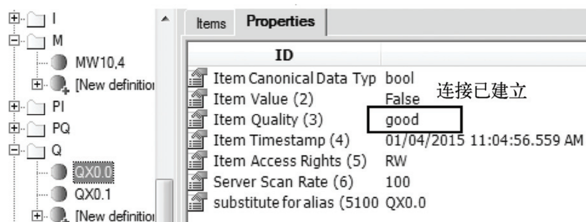


图 13-48　变量的通信状态

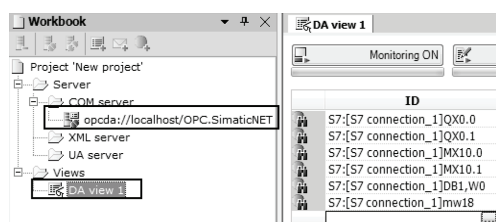


图 13-49　建立变量观察窗口

ID	...	Type	Ac...	Time stamp (U...	Value	Quality
S7:[S7 connection_1]QX0.0	-	bool	RW	01/05/2015 02:06:	True	good
S7:[S7 connection_1]QX0.1	-	bool	RW	01/05/2015 02:06:	True	good
S7:[S7 connection_1]MX10.0	-	bool	RW	01/05/2015 02:06:	False	good
S7:[S7 connection_1]MX10.1	-	bool	RW	01/05/2015 02:06:	False	good
S7:[S7 connection_1]DB1,W0	-	ushort	RW	01/05/2015 02:06:	8195	good
S7:[S7 connection_1]mw18	-	ushort	RW	01/05/2015 02:06:	3	good

图 13-50　变量状态的监控

13.4.5　WinCC 与 OPC 服务器的连接

WinCC 项目可以通过 OPC.chn 获得本机或网络上计算机的 OPC Server 的数据， 操作步骤如下：

在变量管理器中添加 OPC 驱动，如图 13-51 所示，右键单击“OPC Groups”，选择“系统参数”，弹出如图 13-52 所示界面，按照图中的步骤操作，弹出如图 13-53 所示界面，依次添加变量，直到完成所有通信变量的添加，如图 13-54 所示。



图 13-51　选择“系统参数”选项

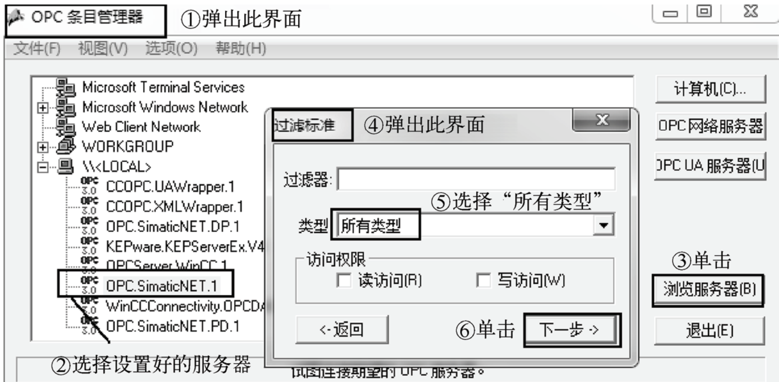


图 13-52　服务器的连接

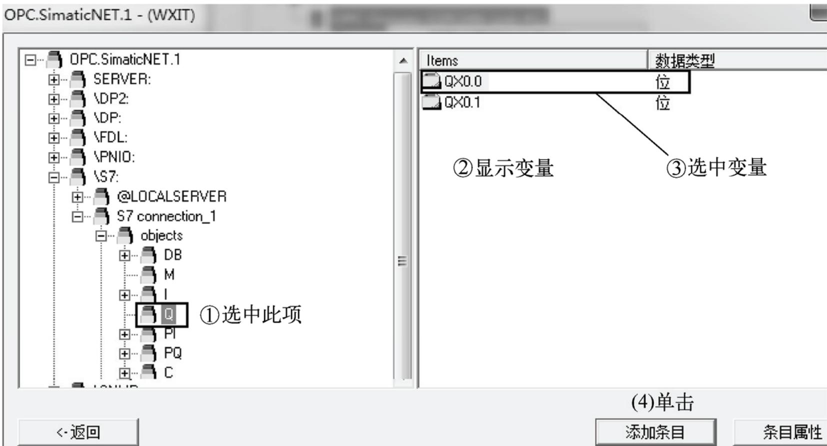


图 13-53　添加变量

WINCC_OPC
计算机
变量管理
内部变量
OPC
OPC Groups (OPCHN Uni
OPC_SimaticNET_1

名称	类型	参数
QX0_0	二进制变量	"S7:[S7 connection_1]QX0.0", "", 11
QX0_1	二进制变量	"S7:[S7 connection_1]QX0.1", "", 11
DB1_W0	无符号 16 位数	"S7:[S7 connection_1]DB1,W0", "", 18
MX10_0	二进制变量	"S7:[S7 connection_1]MX10.0", "", 11
MX10_1	二进制变量	"S7:[S7 connection_1]MX10.1", "", 11
MW18	无符号 16 位数	"S7:[S7 connection_1]MW18", "", 18

图 13-54　WinCC 中变量的建立

13.4.6　系统功能的实现

根据控制要求，WinCC 操作界面制作与调试界面如图 13-55 所示。



a)　　b)

图13-55　WinCC操作界面制作与调试界面

a) 第一台电动机起动　b) 延时8s第二台电动机起动

第 14 章　机加工自动生产线控制系统集成

14.1　系统方案设计

某一自动生产线加工单元的现场布局如图 14-1 所示。线体由输送辊道、工装夹具、停止器及托盘等组成，在线体周边依加工工序和工艺要求布置有立式加工中心、机器人、清洗装置、桁架机械手以及视觉检测装置等设备。

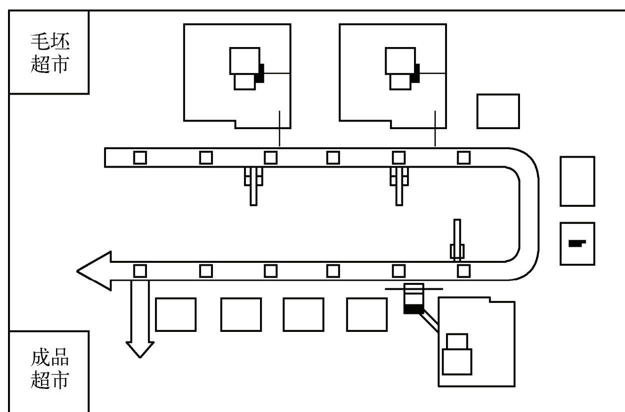


图 14-1　自动生产线布局

整个生产线不仅要求各个机构能够自动配合、加工出合格的产品，而且要求工件从托盘到机床上下料、定位夹紧、机加工以及工件在各工序间的输送、检测等都能自动地进行，为此整个生产线需要通过液压系统、电气控制系统和 PLC 控制系统将各个部分的动作和逻辑关联起来，使其按照设计的程序和预定的节拍自动地工作。

生产单元控制系统结构如图 14-2 所示，其采用“集中管理、分散控制”的控制模式，在 PLC 控制器的作用下，产品的各个工序能有条不紊地、周期性地协调工作。控制系统采用 PROFINET 与 PROFIBUS 混合的网络结构。

PROFINET 和底层的现场 I/O 设备通信，I/O 设备包括 IM151-3PN 现场模块、ET200eco PN 输入/输出模块、RF180C 通信模块等具有以太网功能的模块。为了与车间其他单元数据共享，控制系统还配备了工业级 PN/PN 耦合器，通过该网桥，可以实现自动生产线与车间其他单元如柔性生产单元、立体仓库单元控制器之间的信息交互。

生产线中机床作为专机设备配有 PROFIBUS-DP 模块，机床控制系统、机器人控制系统以及打标机控制系统作为 PLC 的 PROFIBUS-DP 从站接入自动生产线网络。通过总线网络，实现机床、机器人、打标机与 PLC 控制器的双向数字通信，不仅实现了控制中心对机床、机器人和打标机的交互连锁控制，而且实现了实时采集机床、机器人以及打标机的状态信息，例如机器人手爪的状态、机床的主轴速度、运行状态及当前刀具号等信息。

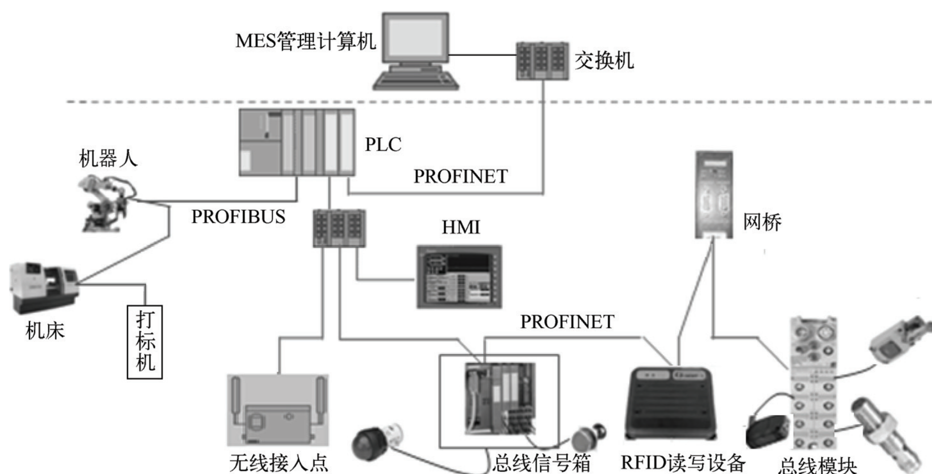


图 14-2　自动生产线控制系统结构

14.2　系统硬件配置及组态

项目结构如图 14-3 所示。自动生产线控制器选择 CPU315-2PN/DP PLC；毛坯超市及成品超市分别安装两个触摸屏（HMI01/HMI02），以便货物出入记录，查询、监控自动生产线工作状态等；系统采用四台伺服电动机，分别用于产品打标系统定位和自动检测系统定位。

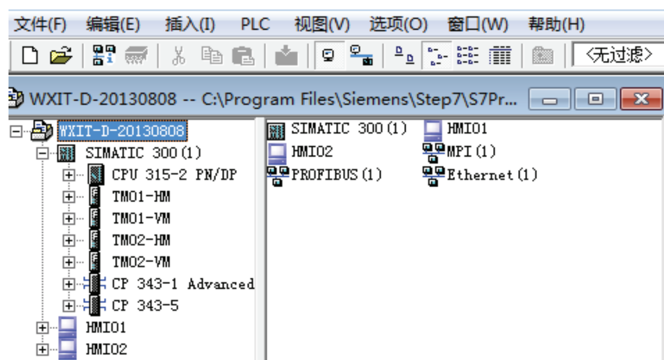


图 14-3　自动生产线项目结构

系统硬件组态界面如图 14-4 所示。

1）机床设备从站。机床配备了 PROFIBUS-DP 接口，在组态前需要安装厂家提供的 GSD 文件，其 DP 属性如图 14-5 所示。

2）IM153-2 接口模块。支持 PROFIBUS 总线通信，可连接 ET200M I/O 分布式设备与 PROFIBUS-DP 设备，最多可操作 12 个 I/O 模块，其配置如图 14-6 所示。

3）IM151-3 PN 高性能接口模块。可连接 ET200S 分布式设备与 PROFINET I/O，可以为装配的电子模块和电动机起动器准备数据，可以为背板总线供电，集成了 2 端口交换机，支持以太网服务：ping、arp、SNMP/MIB-2、LLDP，最多可扩展 63 个模块，部分配置如图 14-7 所示。

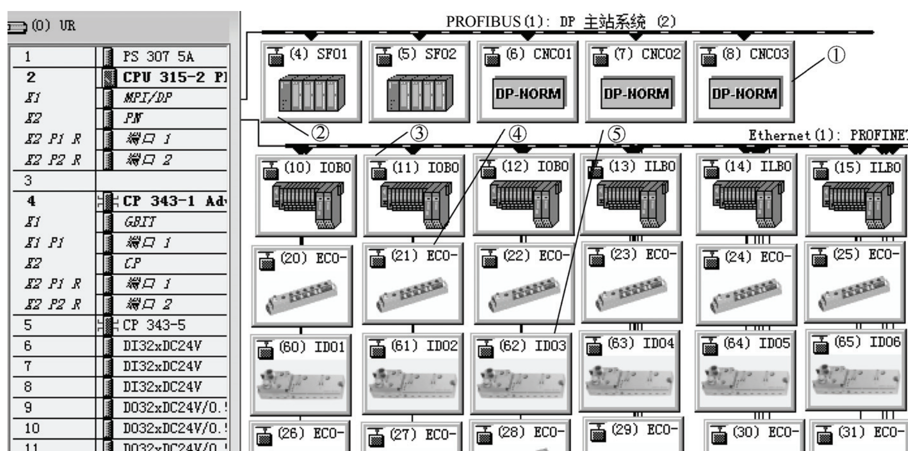


图 14-4　系统硬件组态

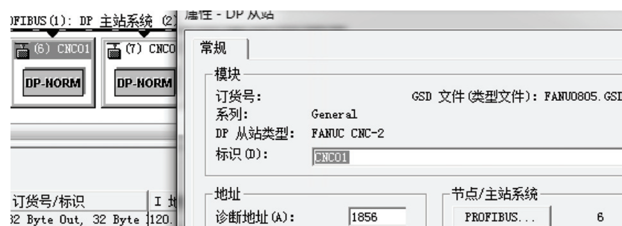


图 14-5　数控机床 DP 从站属性

(4)　SF01

伺服电动机定位模块

插...	模块	订货号	I 地址	Q 地址
1				
2	IM 153-2	6ES7 153-2BA02-0XB0	1859*	
3				
4	TM01-VM	6ES7 353-1AH01-0AE0	312...327	312...327
5	TM01-HM	6ES7 353-1AH01-0AE0	328...343	328...343
6	DI16xDC24V	6ES7 321-1BH02-0AA0	100...101	
7	DO16xDC24V/0.5A	6ES7 322-1BH01-0AA0		100...101
8	CP 340-RS232C	6ES7 340-1AH01-0AE0	392...407	392...407

图 14-6　ET200M 分布式 I/O 系统构成

(10)　IOB01

插...	模块	订货号	I 地址	Q 地址	诊断地址
0	IOB01	6ES7 151-3BA23-0AB0			2040*
X1	PN-IO				2039*
X1 ...	端口 1				2042*
X1 ...	端口 2				2041*
1	PM-E DC24V	6ES7 138-4CA01-0AA0			2017*
2	4DI DC24V ST	6ES7 131-4BD01-0AA0	20.0...20.3		
3	4DI DC24V ST	6ES7 131-4BD01-0AA0	20.4...20.7		1844*
4	4DI DC24V ST	6ES7 131-4BD01-0AA0	21.0...21.3		
5	4DI DC24V ST	6ES7 131-4BD01-0AA0	21.4...21.7		1843*

图 14-7　ET200S 分布式系统

4）ET200ecoPN 是一款紧凑型、防护等级为 IP65/67 的 PROFINET I/O 接口模块，尤其适合在无控制柜的应用场合。ET200ecoPN 模块拥有全密封锌压铸外壳，设计非常紧凑，配置如图 14-8 所示，有 2 路 PROFINET 通信接口、8 路数字量输入信号。

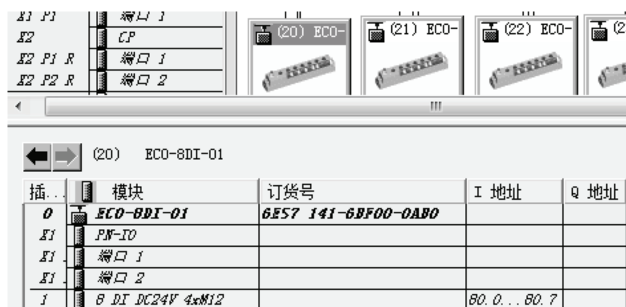


图 14-8　ET200ecoPN 接口模块配置

5）RF180C 通信模块的防护等级为 IP67，可连接两路具有 PROFINET I/O 功能的读写器，配置如图 14-9 所示。

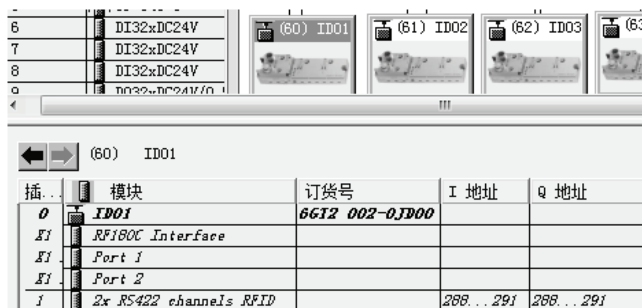


图 14-9　RF180C 通信模块

14.3　RFID 信息识别功能的实现

为了实现对生产线上产品生产信息的全程跟踪、实时记录和有效追溯及对产品的库存管理，在生产线上的每一道工序都布置有 RFID 读写器 RF380R，对应的电子标签 RF340T 安装在承载物料的托盘上，从而形成生产管理系统与现场生产信息的连接通道。

通信处理器选用 RF180C PN，每个处理器可以连接两台读写装置。读写器通过工业以太网接口连接到 RF180C 模块上，RF180C 模块通过工业以太网接口连接到其他以太网设备接口上，其硬件组态方式如图 14-10 所示。

读写器采集生产数据，由 PLC 将采集到的状态数据发送到上位机并接收上位机指令，上位机对生产过程进行统一调度和监控。

图 14-11 所示为读写器往电子标签 MDS 中写入信息的编程软件监控界面，[DB47.DBB0]=1 为写指令（为 2 是读指令），用于将存储在［DB47.DBB6］指定的数据块数据写入 MDS 中，写入信息长度为［DB47.DBB2］中设定的字节长度值。

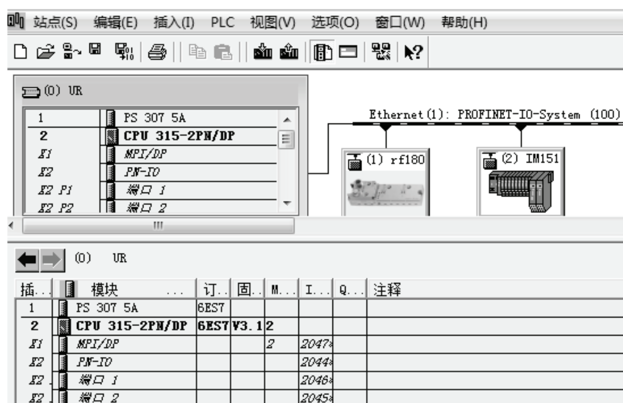


图 14-10　RF180C 的硬件组态

	地址	符号	状态值	修改数值
1	// Command Start			
2	DB45.DBX 19.1	"MOBY DB".SLG1.command_start	false	true
3	// Ready			
4	DB45.DBX 18.7	"MOBY DB".SLG1.ready	true	
5	// Presence of a MDS			
6	DB45.DBX 18.0	"MOBY DB".SLG1.ANZ_MDS_present	true	
7	// MOBY Command			
8	DB47.DBB 0	"Command".Kanal_1_Befehl[1].command	B#16#01	B#16#01
9	DB47.DBB 1	"Command".Kanal_1_Befehl[1].sub_command	B#16#00	
10	DB47.DBW 2	"Command".Kanal_1_Befehl[1].length	2	2
11	DB47.DBW 4	"Command".Kanal_1_Befehl[1].address_MDS	W#16#0000	
12	DB47.DBW 6	"Command".Kanal_1_Befehl[1].DAT_DB_number	48	48
13	DB47.DBW 8	"Command".Kanal_1_Befehl[1].DAT_DB_address	W#16#0000	
14	DB50.DBB 0		B#16#00	
15	DB50.DBB 1		B#16#10	

图 14-11　写入 MDS 信息

通过传感器、控制器及 MES 的配合，将工序信息写入电子标签，或将电子标签信息传送到 MES 系统，MES 系统对数据进行分析转发，实现对产品信息的识别、跟踪、查询和追溯。图 14-12 所示为 MES 系统对自动生产线某一产品的跟踪，其中事件 MOVE 是指移入操作，当前加工件从上一步初加工进入自动检测工序，进站时间是 13 点 25 分。

事件日期	事件码	工作流节点名称	
2014年03月31日 13:25	MOVE	W50-自动检验 1.0	OPCClient

数量	1
上一步工作流节点	W20-初步加工 1.0
当前工作流节点	W50-自动检验 1.0
入站时间	2014年03月31日 13:25
入站操作员	OPCClient
入站接受资源	ID03
本工序累耗时(秒)	0
批号发出时间	2014年03月31日 13:25
出站操作员	OPCClient

图 14-12　加工件进入下一工序动作跟踪

图 14-13 所示为 HMI 上监控的第七个工位上的 RFID 信息，实时监控当前 RFID 命令与读/写状态、标签信息等。



图 14-13　RFID 状态监控界面

可见，基于 RFID 技术的现代物流系统，有利于增强工厂管理水平，提高生产效率，同时也是工厂向现代化制造业转变、融入现代物联网的重要部分。

14.4　打标系统的集成

1．系统分析

在生产线末端设有二维条码机打码工位，通过 RFID 读/写器读/写电子标签内物料信息，并将信息发送到打标服务器，如果产品检验合格则启动打标机工作程序，完成对产品的打标工作，使得系统能对加工好的成品进行信息追溯和管理，如果产品检验不合格则直接放行进入废品收纳通道。

根据控制需求，产品打标位置应精确定位，因此桁架机械手的 X/Y 方向分别用伺服电动机定位，需要用到定位模块 FM353，FM353 通过远程 I/O 模块 ET200M 接入主控制器系统，PLC 作为 DP 主站与分布式 I/O 模块交换数据并监控 PROFIBUS-DP 网络。

2．远程 I/O 站配置

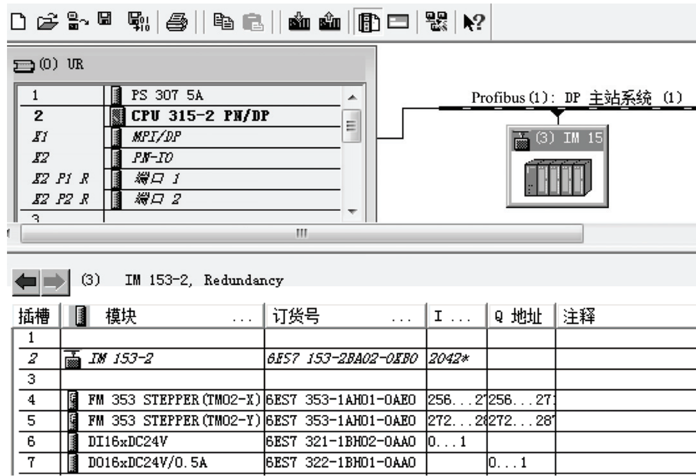
ET200M 模块配置如图 14-14 所示，模块的 PROFIBUS-DP 接口通过 PROFIBUS-DP 电缆与 PROFIBUS-DP 主站 CPU315-2PN/DP 的 DP 接口连接。远程 I/O 站按照 PS307 模块 + IM153-2 模块 + FM353 模块（X 方向）+ FM353（Y 方向）+ DI16*DC24V 模块 + DO16*DC24V/0.5A 模块的顺序安装在与 S7-300 PLC 相同的导轨上。



图 14-14　ET200M 的配置

3．PROFIBUS-DP 从站配置

系统中从站模块是 ET200M 模块，机架上分别插入 IM153-2、FM353（X 轴）、FM353（Y 轴）以及 DI 模块、DO 模块，如图 14-15 所示。



插槽	模块	订货号	I ...	Q 地址	注释
1					
2	IM 153-2	6ES7 153-2BA02-0XB0	2042*		
3					
4	FM 353 STEPPER(TM02-X)	6ES7 353-1AH01-0AE0	256...2	256...27	
5	FM 353 STEPPER(TM02-Y)	6ES7 353-1AH01-0AE0	272...2	272...28	
6	DI16xDC24V	6ES7 321-1BH02-0AA0	0...1		
7	DO16xDC24V/0.5A	6ES7 322-1BH01-0AA0		0...1	

图 14-15　ET200M 硬件组态

4．定位模块 FM353 功能实现

FM353 模块与伺服驱动器通信信号包括正负脉冲输出、正负旋转方向输出等（模块驱动器接口信号中的位置脉冲、方向、驱动使能以及电流调节输出信号均为差分输出）。FM353 将脉冲信号和方向信号发送到伺服驱动器中，从而控制伺服电动机运动，完成打标工件的定位任务。FM353 共有 7 种工作模式，如图 14-16 所示。

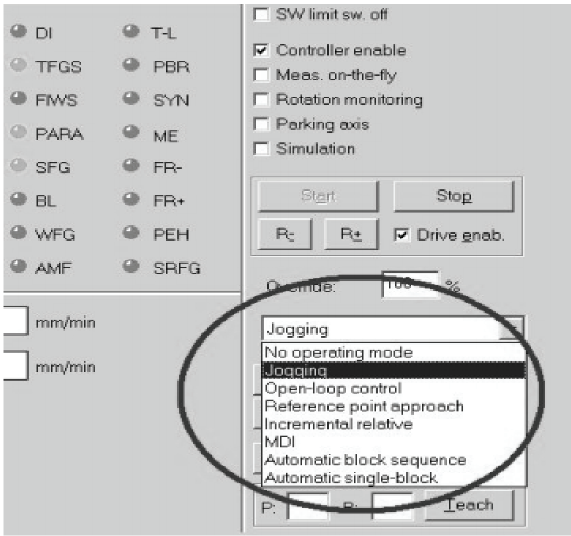


图 14-16　伺服电动机工作模式的选择

打标运动系统设置了三种模式，分别是 Jogging、Reference point approach 以及 MDI 模式。

Jogging 模式是一种利用外部的控制按钮移动的控制方式，需要外部提供运动方向信号。

Reference point approach 模式用于消除 FM353 的零点偏置，为了定位水平和垂直的原点，

在桁架机械手的水平轴和垂直轴上分别安装了档块，在电动机上安装有接近开关，用于确定系统的参考点。

MDI 模式是一种半自动工作模式，用于自动执行手动输入的数据。

桁架机械手 HMI 坐标设定界面实现如图 14-17 所示，电动机运行的步骤显示如图 14-18 所示。

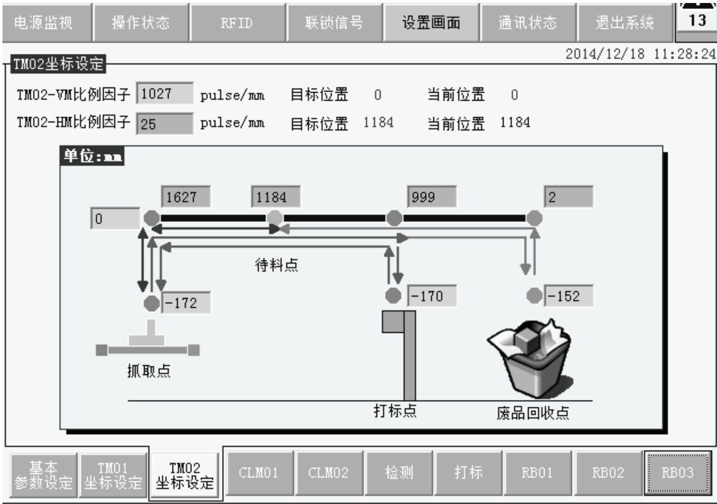


图 14-17　坐标值设定



图 14-18　步进电动机状态监控

5．打标系统调试

系统调试包括硬件调试、软件调试及系统联调。硬件调试主要是现场伺服驱动器参数设置、参考点定位以及工件能否正常抓取/夹紧等工作；软件调试主要是通过测试点动、

寻找参考点、半周期运行等工作模式调整系统逻辑关系是否符合运行要求；系统联调则是将现场全部设备配合起来联动、调节功能是否正常、工件打标位置是否符合要求以及生产现场是否符合要求等。图 14-19 所示为桁架机械手的 Y 轴下降至自动线上工件指定抓取位置，并夹紧工件；图 14-20 所示为桁架机械手通过 X、Y 轴的运动将工件运送到打标指定位置进行激光打码。

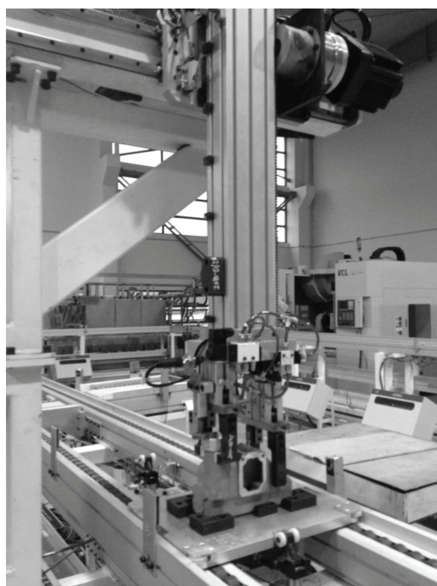


图 14-19　机械手在线上抓取工件

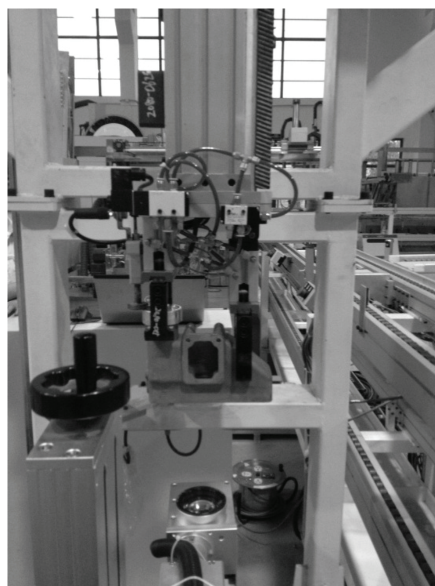


图 14-20　机械手将工件放在打标位置

14.5　MES 系统与 PCS 系统的集成

采用 OPC 技术实现上位机 MES 与 PLC 之间的通信。PLC 系统和上位机之间需要建立一定的应答机制，以便上位机采集现场数据、跟踪现场信息，依据跟踪结果判断生产工艺参数的合理性，并下发产品动作指令，使得 MES 系统能对工艺路线控制信息、现场设备故障信息、现场工位缺料信息、现场订单执行进度信息以及现场设备操作人员信息等基础数据进行管理。

图 14-21 所示为 MES 系统与 PLC 系统建立的应答机制，该数据通过 RFID 读写器获取并存入指定的数据块 DB 中。当 MES 计算机成功获取某一工序数据后会给 PLC 反馈信号“1”，托盘进入下一工序。

101.0	MES_ID03.MES_To_Au	BYTE	B#16#00	MES计算机成功获取数据后，给PLC反馈1信号
102.0	MES_ID03.Spare2	BYTE	B#16#00	备用
103.0	MES_ID03.Spare3	BYTE	B#16#00	备用
104.0	MES_ID03.Pallet_No	BYTE	B#16#30	托盘编号
105.0	MES_ID03.Pallet_No	BYTE	B#16#32	托盘编号
106.0	MES_ID03.Pallet_No	BYTE	B#16#39	托盘编号
107.0	MES_ID03.Pallet_St	BYTE	B#16#00	托盘状态
108.0	MES_ID03.Materialc	BYTE	B#16#31	物料编码
109.0	MES_ID03.Materialc	BYTE	B#16#39	物料编码
110.0	MES_ID03.Materialc	BYTE	B#16#30	物料编码

图 14-21　MES 通过 RFID 读写器读取的信息

图 14-22 所示为 MES 对物料的管理信息，通过对 RFID 信息的读/写，自动记录加工件工序号、工序实际开始和结束时间、工序具体加工的设备/工位以及操作者等生产信息。

事件日期	工作流节点名称	用户名称
2014年03月31日 13:31	W50-自动检验 1.0	OPCClient

工作中心描述	资源名称	资源描述
自动线	ID05	RFID读卡器'ID05',用于三坐标自动检测仪（CMM）

事件日期	工作流节点名称	用户名称
2014年03月31日 13:32	W50-自动检验 1.0	OPCClient
2014年03月31日 13:32	W50-自动检验 1.0	OPCClient
2014年03月31日 13:32	W60-二次加工 1.0	OPCClient
2014年03月31日 13:34	W60-二次加工 1.0	OPCClient

图 14-22　加工件在生产线上各个工序时间节点

14.6　车间级物料小车控制系统集成

自动生产线进出口处设有毛坯超市和成品超市。毛坯超市在生产线的入口，成品超市在生产线的出口。在超市的辊道入口处设有三个按钮，分别是小车呼叫（缺料呼叫/成品入库呼叫）、校验按钮和确认按钮，其中校验按钮用于对毛坯的校验，当小车上运载的货物与生产单元请求货物批号不一致时，MES 发出错误信号，生产单元毛坯辊道线上的停止器关闭不放行，货物不能进入生产单元。

如图 14-23 所示，工厂智能管控平台作为无线接入点即 AP 主站，采用 SCALANCE W788-1RR 模块并配有 2.4GHz 全向天线，其无线信号可以覆盖整个生产车间；车载控制系统接收端采用 SCALANCE W747-1RR 客户端，实现物料小车在车间内的漫游功能，接收来自 AP 主站的信息，然后显示在车载显示界面，实现了人与车之间的对话，并发出不同的语音提示信息，小车作业人员依据车载显示界面的导航信息操作物料小车，从而提高了物料小车的利用效率、缩短了物料小车的调度时间并大大减少了货物送错的概率。

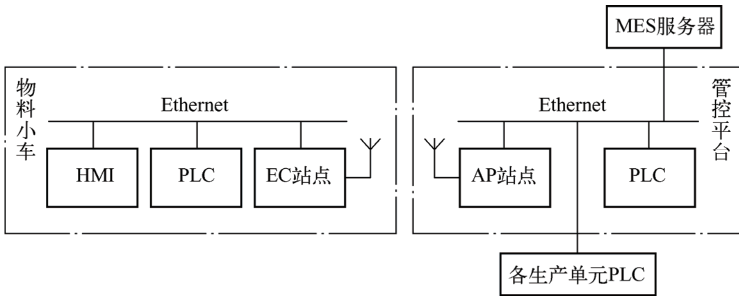


图 14-23　物料小车与车间控制系统的集成

当自动生产线有呼叫信号时，管控平台接收该信息并处理，然后通过无线网络向物料小车发送，物料小车最终成功接收到来自自动线的呼叫信息、作出判断并在小车 HMI 界面显示相关信息，提示小车操作人员驶向立体仓库领取毛坯并运送毛坯至生产线毛坯超市或者驶向生产线成品超市执行入库任务。

图 14-24~图 14-28 所示为自动线与物料小车信息交互数据表及程序，通信采用 FB12/FB13 模块实现。

图 14-24 所示为 DB2102 内容，是自动线与物料小车信息交互的内容，包括缺料、成品

入库呼叫、小车 HMI 显示的物料信息以及通信状况显示等内容。

地址	名称	类型	初始值	注释
0.0		STRUCT		
+0.0	lack_of_matria	BOOL	FALSE	缺料信息
+0.1	send_to_A	BOOL	FALSE	成品发货
+0.2	Mteria_CLr_inf	BOOL	FALSE	货物信息清零可
+0.3	Blank_sure_inf	BOOL	FALSE	毛坯信息确认
+0.4	End_sur_inf	BOOL	FALSE	调温器体成品信息确认
+1.0	T_BOX_code_BS	BYTE	B#16#0	毛坯超周转筐编号
+2.0	T_BOX_code_BS1	BYTE	B#16#0	毛坯超周转筐编号
+3.0	T_BOX_code_BS2	BYTE	B#16#0	毛坯超周转筐编号
+4.0	spare	BYTE	B#16#0	备用
+5.0	T_BOX_code_PS	BYTE	B#16#0	成品超市周转筐编号
+6.0	T_BOX_code_PS1	BYTE	B#16#0	成品超市周转筐编号
+7.0	T_BOX_code_PS2	BYTE	B#16#0	成品超市周转筐编号
+8.0	Materialcode_PS	BYTE	B#16#0	成品超市物料编码
+9.0	Materialcode_PS1	BYTE	B#16#0	成品超市物料编码
+10.0	Materialcode_PS2	BYTE	B#16#0	成品超市物料编码
+11.0	Materialcode_PS3	BYTE	B#16#0	成品超市物料编码
+12.0	Materialcode_PS4	BYTE	B#16#0	成品超市物料编码
+13.0	Materialcode_PS5	BYTE	B#16#0	成品超市物料编码
+14.0	Materialcode_PS6	BYTE	B#16#0	成品超市物料编码
+15.0	Materialcode_PS7	BYTE	B#16#0	成品超市物料编码
+16.0	Materialcode_PS8	BYTE	B#16#0	成品超市物料编码
+17.0	Materialcode_PS9	BYTE	B#16#0	成品超市物料编码
+18.0	Materialcode_PS10	BYTE	B#16#0	成品超市物料编码
+19.0	Material_Number_PS	BYTE	B#16#0	成品超市物料数量
+20.0	Material_Number_PS1	BYTE	B#16#0	成品超市物料数量
+21.0	Material_Number_PS2	BYTE	B#16#0	成品超市物料数量
+22.0	Material_Number_PS3	BYTE	B#16#0	成品超市物料数量
+23.0	Beat	BOOL	FALSE	通信信号

图 14-24　自动生产线产品信息

图 14-25 所示为自动线向管控平台发送数据的程序，包括备用字节共 28 个字节，发送模块 FB12 的 R_ID 包含了远程设备的地址信息身份。

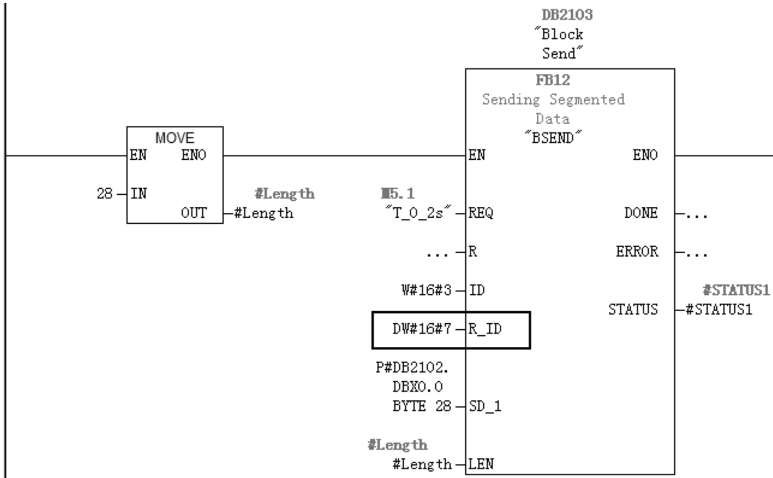


图 14-25　自动线向管控平台发送数据

图 14-26 所示为管控平台接收自动生产线数据，并将数据传送到 DB320 数据块中，FB13 通信模块的接收地址 R_ID 要与图 14-25 的 R_ID 编号保持一致，使得这两个模块之间形成通信的对应关系。

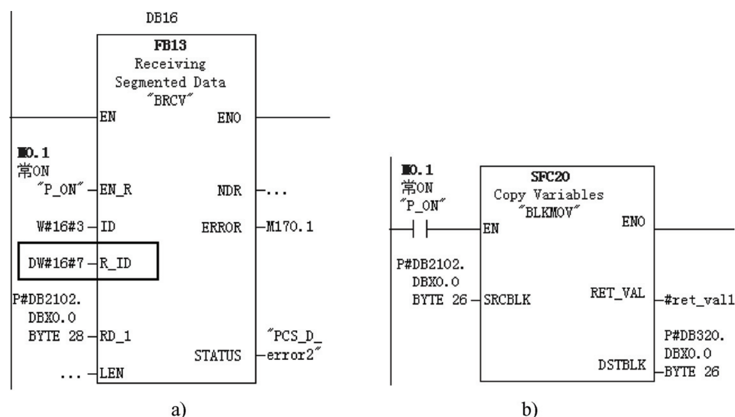


图 14-26　管控平台接收并存储自动线数据

a) 接收数据　b) 转移数据

图 14-27 所示为管控平台向物料小车发送数据的一段程序，DB320 为管控平台接收的来自自动生产线毛坯产品信息，通过 FB12 通信模块将此信息发到物料小车的 PLC 中，以便小车上的 HMI 显示此信息。

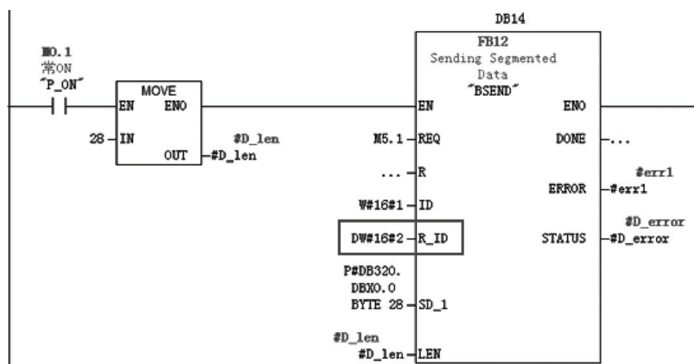


图 14-27　管控平台向物料小车发送数据

图 14-28 所示为物料小车接收管控平台发送的自动生产线毛坯产品信息，通过 FB13 接收模块接收信息，其中 R_ID 要与图 14-27 的 R_ID 编号保持一致，使得这两个模块之间形成通信的对应关系。

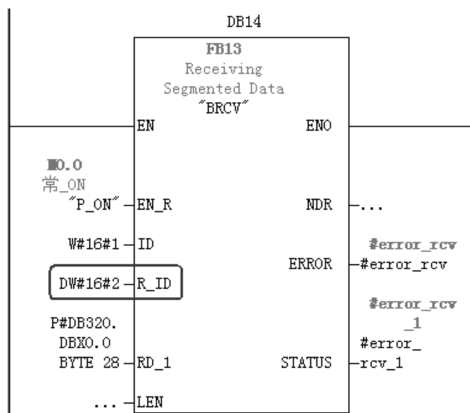


图 14-28　物料小车接收管控平台数据

当小车上装载物料时，车载显示界面将同步实时显示物料信息，如图 14-29 所示，包含筐号、物料名称、物料代码以及当前去向等信息。

图 14-29　车载物料信息

14.7　项目总结

智能工厂机加工生产线采用 PROFINET 和 PROFIBUS 混合的控制网络。其中 PROFINET 网络利用网络交换机构成星型拓扑结构，组网方式非常灵活，可以在配电柜及被控设备附近布置远程 I/O 站，设备之间采用专用电缆连接，这种网络结构节省电缆、维护方便，且能与 MES 等其他系统实现无缝对接；为了保证底层设备运行的可靠性和快速性，底层关键设备采用 PROFIBUS 网络通信。

机加工生产线经过现场调试，各方面运行状况良好，产品加工精度符合要求；不仅实现了生产线上多设备信息采集和共享，而且大大提高了设备的安全性、生产过程的透明性和企业管理的有效性。

参 考 文 献

[1] 郭琼. PLC 应用技术 [M]. 2 版. 北京：机械工业出版社，2013.

[2] 西门子（中国）有限公司自动化与驱动集团. 深入浅出西门子 S7-200 PLC [M]. 北京：北京航空航天大学出版社，2003.

[3] SIEMENS. SIMATIC STEP7 V5.4 编程使用手册. 2006.

[4] SIEMENS. SIMATIC S7-300 CPU31xC 和 CPU31x 技术数据手册. 2006.

[5] SIEMENS. SIMATIC S7-300 可编程序控制器硬件和安装手册. 2002.

[6] SIEMENS. SIMATIC S7-300 和 S7-400 的梯形图（LAD）编程. 2004.

[7] SIEMENS. SIMATIC 用于 S7300/400 系统和标准功能的系统软件—卷 1/2. 2007.

[8] SIEMENS. SIMATIC 分布式 I/O 系统 ET200S 操作说明. 2005.

[9] SIEMENS. SIMATIC 分布式 I/O 设备 ET200eco 操作指导. 2006.

[10] 郭琼. 现场总线技术及其应用 [M]. 2 版. 北京：机械工业出版社，2013.

[11] 郭琼，姚晓宁. 机加工自动生产线控制系统集成 [J]. 制造业自动化，2014,36(8):16~18,29.

[12] 刘齐忠，吴文信，孙晓晓. 以 DCS 为基础的过程控制系统集成[J]. 石油化工自动化，2010,6:36~39.

[13] 朱晓梅，韩红军. 基于 XML 和 OPC 的工业控制系统集成的研究 [J]. 中国科技信息，2006.2:11,14

在线互动交流平台

官方微博：http://weibo.com/cmpjsj
豆 瓣 网：http://site.douban.com/139085/
读者信箱：cmp_itbook@163.com

S7-200/S7-300 PLC 基础及系统集成







本书特色

- 紧贴控制技术的新发展和新应用，内容先进和实用；
- 以案例带动知识点组织内容，理论以够用为度，案例具体翔实；
- 本书涉及STEP7、Wincc flexible、WinCC及SIMATIC NET等西门子常用软件；以及S7-200 PLC、S7-300 PLC、MM4变频器等硬件。

地址：北京市百万庄大街22号
邮政编码：100037
电话服务
服务咨询热线：010-88361066
读者购书热线：010-68326294
010-88379203
网络服务
机 工 官 网：www.cmpbook.com
机 工 官 博：weibo.com/cmp1952
教育服务网：www.cmpedu.com
金 书 网：www.golden-book.com
封面无防伪标均为盗版



机械工业出版社
微信公众号



计算机分社微信服务号

上架指导 工业技术/PLC

ISBN 978-7-111-51420-6

策划编辑◎ 时 静 /封面设计◎





ISBN 978-7-111-51420-6
9 787111 514206
定价：59.80元